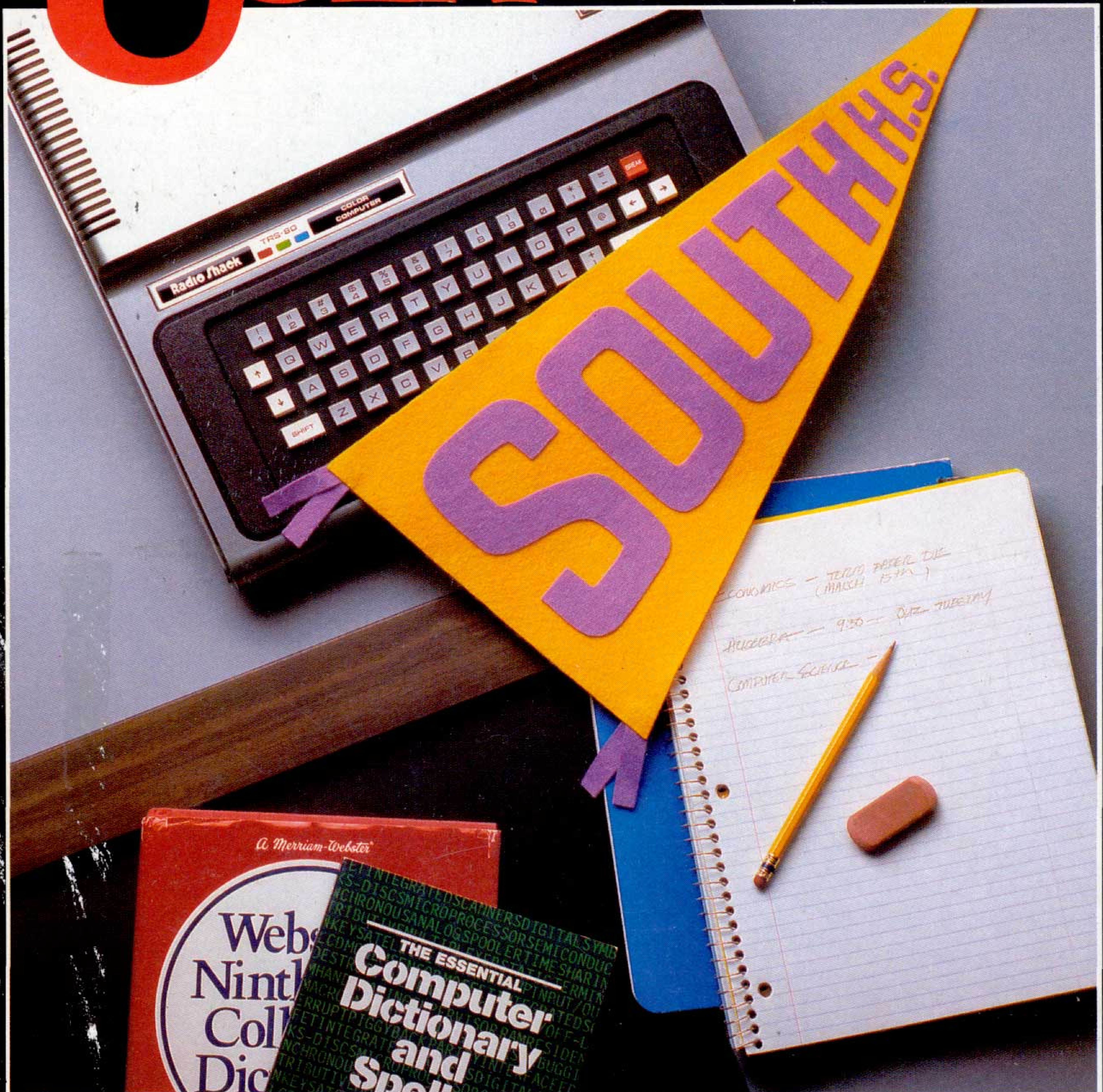
For the Tandy/Radio Shack System

COMPUTER USER™

February 1984 $2.95 Canada $3.50



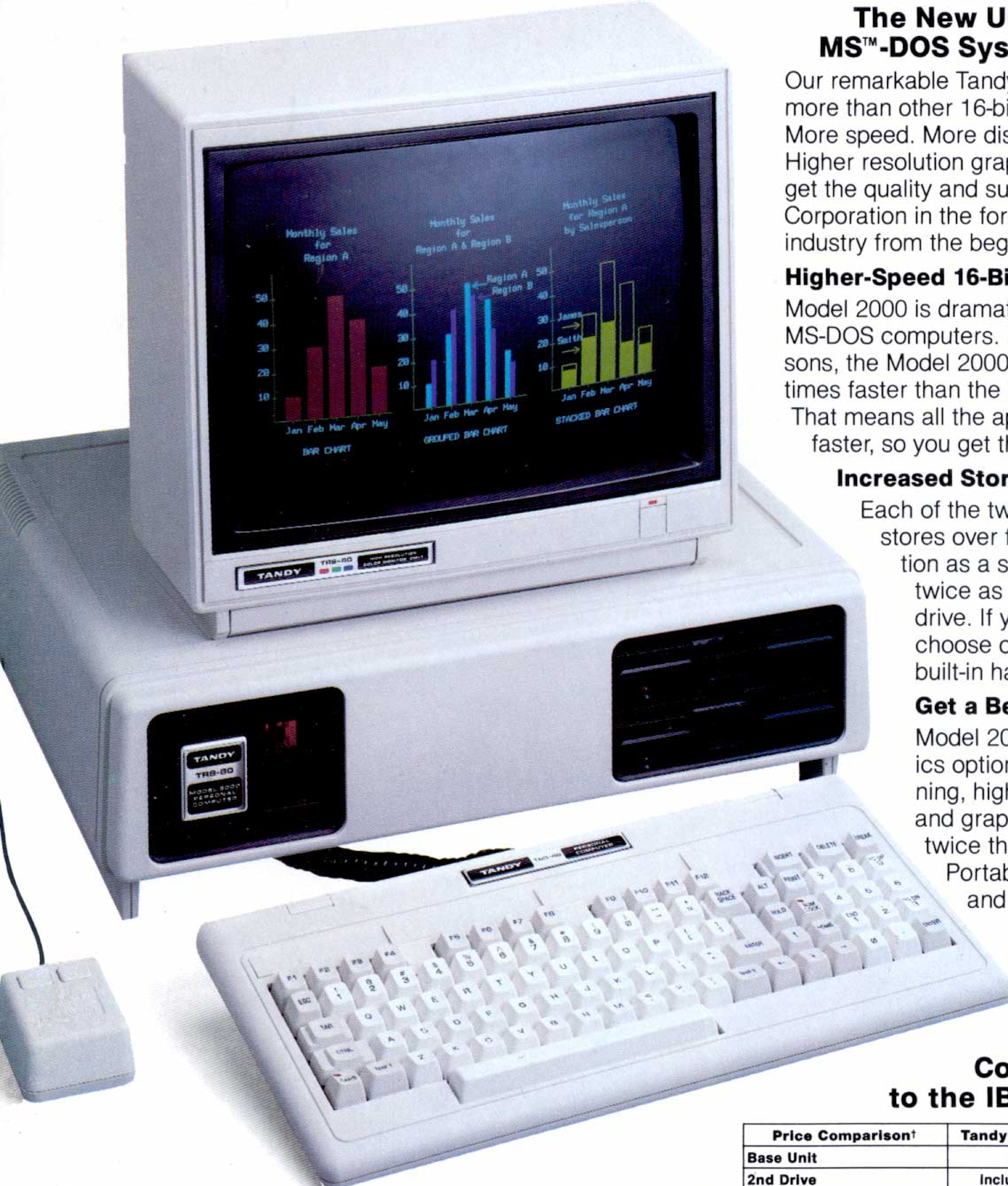
- Preschool to College: Computerizing Education
- How to Build a Super Micro
- Apple IIe vs Model 4: And the Winner is. . .
- Zapping Your CoCo

02

0 71486 02857

02857

THE TANDY TRS-80® MODEL



The New Ultra-High Performance MS™-DOS System from Radio Shack

Our remarkable Tandy Model 2000 delivers much more than other 16-bit MS-DOS based computers. More speed. More disk storage. More expansion. Higher resolution graphics. But that's not all. You get the quality and support that have kept Tandy Corporation in the forefront of the microcomputer industry from the beginning.

Higher-Speed 16-Bit CPU

Model 2000 is dramatically faster than other 16-bit MS-DOS computers. In actual benchmark comparisons, the Model 2000 performed almost three times faster than the IBM® Personal Computer. That means all the applications software runs faster, so you get the job done quicker.

Increased Storage Capacity

Each of the two Model 2000 disk drives stores over four times as much information as a standard IBM PC drive, and twice as much as a TI® Professional drive. If you need even more storage, choose our Model 2000 HD with a built-in hard disk.

Get a Better Picture of Things

Model 2000 video monitors and graphics options let you easily create stunning, highly-detailed business charts and graphs. Our color graphics have twice the resolution of the COMPAQ™ Portable or IBM PC color graphics, and twice the number of colors.

TRS-80 Model 2000 With Two Disk Drives

2750.00

Commerical Lease Available For Only $95 Per Month (Plus Applicable Use/Sales Tax)

TRS-80 Model 2000 HD With Built-In 10-Megabyte Hard Disk

4250.00

Commercial Lease Available For Only $150 Per Month (Plus Applicable Use/Sales Tax)

Monitor, graphics, Floor stand, Pedestal and Digi™-Mouse not included

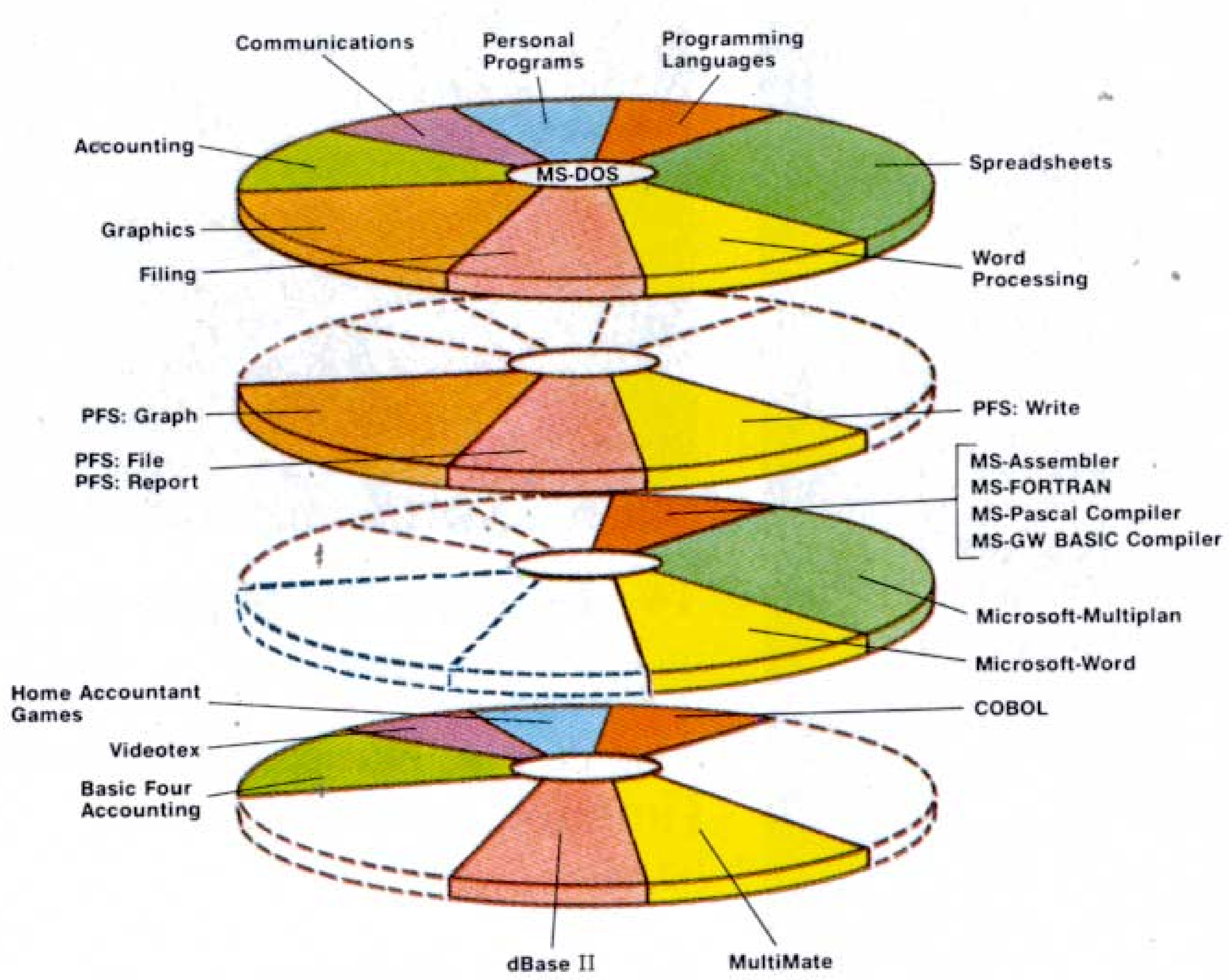
Compare the Model 2000 to the IBM Personal Computer

Price Comparison†	Tandy Model 2000	IBM Personal Computer
Base Unit	$2750	$2104
2nd Drive	Included (720K)	$529 (320K)
Monochrome Monitor	$249	$345
Display/Printer Adapter	Included	$335
128K RAM	Included	$165
RS-232	Included	$120
MS-DOS 2.0	Included	$60
Total Cost*	$2999	$3658
Feature Description	**Tandy Model 2000**	**IBM Personal Computer**
Internal Memory	128K Standard	64K Standard
Disk Capacity Per Drive	720K	160K or 320K (optional)
Microprocessor Clock Speed	8 MHz	4.7 MHz
True 16-Bit Microprocessor	Yes (80186) 16 bit/16 bit data path	No (8088) 16 bit/8 bit data path
User-Available Expansion Slots*	4	2
Grapics Options		
Color Resolution	640 × 400	320 × 200
Number of Colors	8	4
Monochrome Resolution	640 × 400	640 × 200

*Comparable IBM configuration with monochrome adapter and display, communications adapter, two 320K disk drives and 128K RAM. †Manufacturer's pricing as of 9/1/83.

2000 PERSONAL COMPUTER

Model 2000 Adds a New Dimension to the Most Advanced MS-DOS Software Around



Choose From the Hottest Names In Software

Rapidly becoming an industry standard, the MS-DOS operating system lets us bring you the most popular programs around. Programs like dBase II™. Microsoft-Multiplan™. PFS®:File. MultiMate™. And many more already acclaimed by the entire microcomputer industry. The four software "wheels" above illustrate the outstanding programs available. Each "slice" in the top wheel represents typical applications—electronic filing, word processing, spreadsheets, communications and more. The corresponding slices on the lower wheels portray the programs available for each application family. For example, we have an excellent line of small business accounting software from MAI/Basic Four™—general ledger, accounts payable, accounts receivable and more. These flexible programs have already been proven in thousands of applications worldwide. More programs are under development, including Microsoft's state-of-the-art MS-Windows operating environment. You can view several "windows" at once—files, letters, graphs and more—and exchange information between them.

Radio Shack®

The biggest name in little computers®

A DIVISION OF TANDY CORPORATION

Room to Grow for the Future

Model 2000 has many standard features that cost extra on other computers. Aside from obvious cost advantages, that leaves more expansion slots open for your future needs. Add additional memory, a hard disk drive, our Digi™-Mouse and more. And you can install most options yourself.

A Milestone in Ergonomic Design

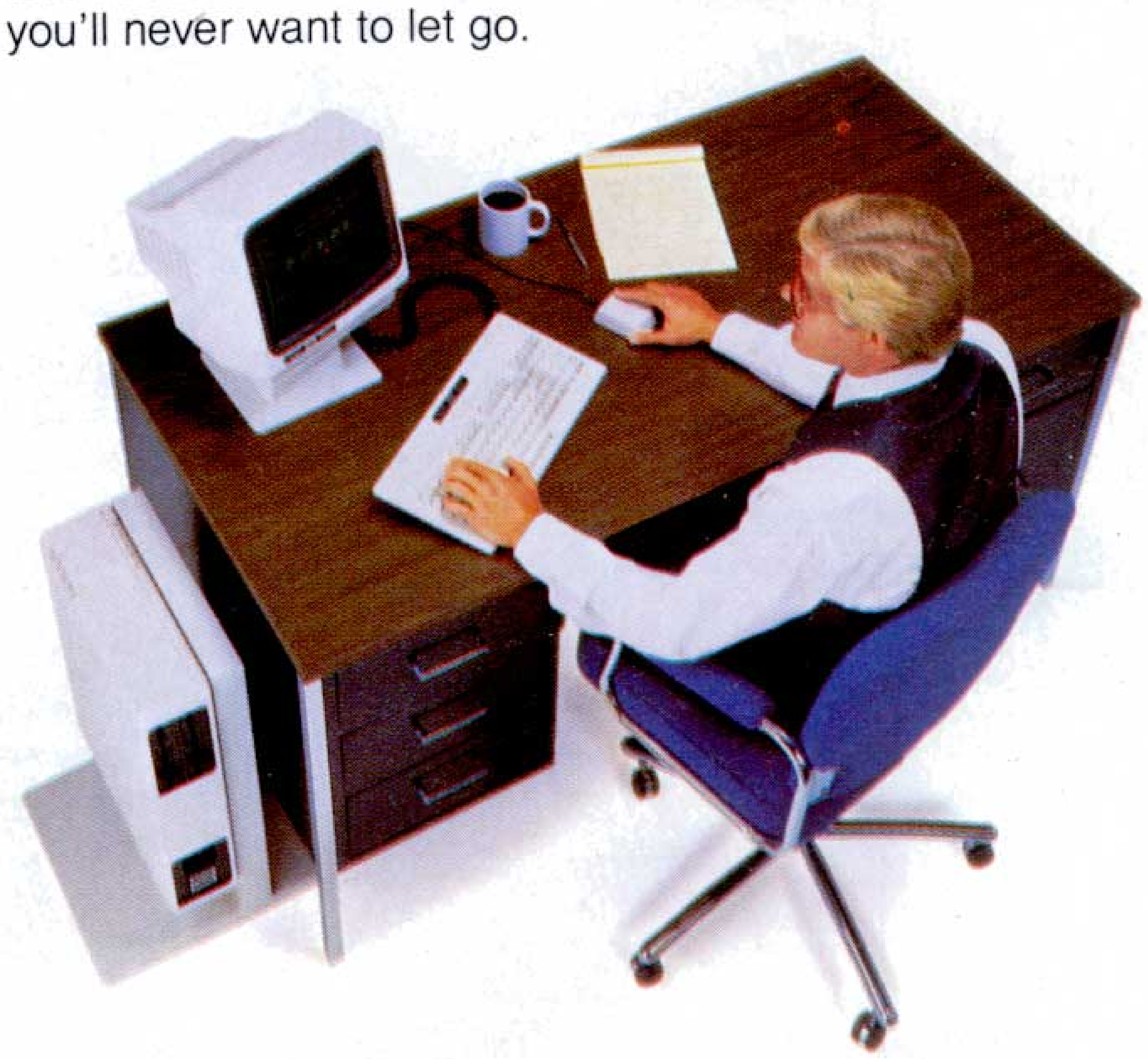
Model 2000 is as spectacular outside as it is inside. The attractive modular system adapts to your environment to assure maximum comfort and productivity. With the optional floor stand, Model 2000 occupies much less desk space than other personal computers.

Total Service and Support is Close at Hand

Our extensive network of Radio Shack Computer Centers gives you complete hardware and software support directly from the manufacturer, not an independent computer dealer. On-site and carry-in service is available nationwide.

The Future of Personal Computing

See the TRS-80 Model 2000 and its popular line of software today at over 1000 Radio Shack Computer Centers nationwide. Ask for a hands-on demonstration and you'll never want to let go.



Send me the TRS-80 Model 2000 brochure.

Mail To: Radio Shack, Dept. 84-A-786
300 One Tandy Center, Fort Worth, Texas 76102

NAME ____________________

COMPANY ____________________

ADDRESS ____________________

CITY ____________ STATE ______ ZIP ______

TELEPHONE ____________________

CIRCLE NO. 47 ON INQUIRY CARD

For the Tandy/Radio Shack System

COMPUTER USER™

VOLUME 2 ISSUE 2 FEBRUARY 1984

Distributed in the U.S. and
Canada by
Curtis Circulation Co.
West Caldwell, NJ

International Newsstand Distribution
Director Lew Ullian
Oberstr. 38,
6000 Frankfurt 61
West Germany
Telephone: (0611) 41 84 80

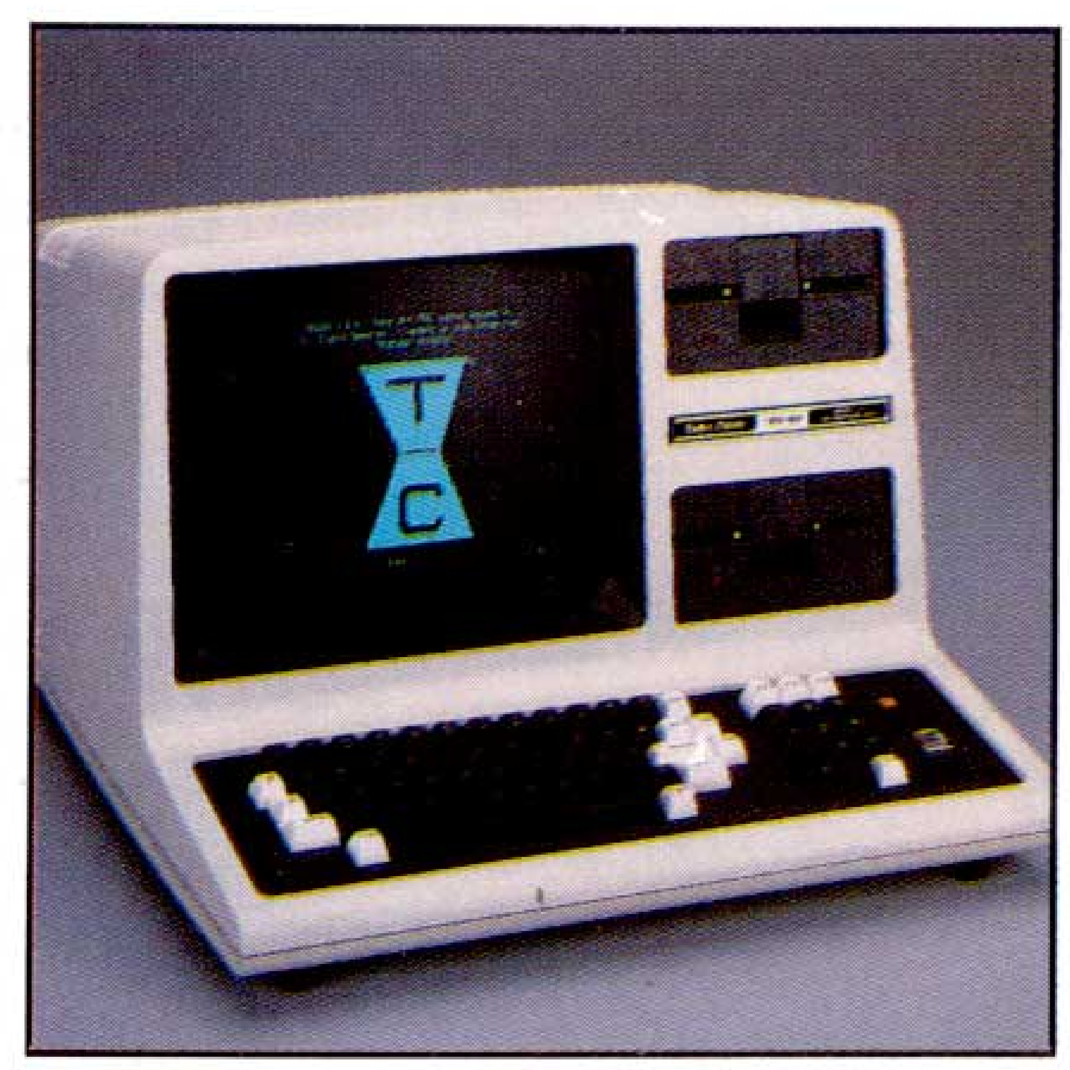
Main Office
16704 Marquardt Avenue
Cerritos, CA 90701
(213) 926-9544
TWX (910) 583-1412

Southwest Regional Office
Tandy Center Atrium #550
Fort Worth, TX 76102
(817) 335-6245

COMPUTER USER Magazine, published monthly by McPheters, Wolfe & Jones, 16704 Marquardt Ave., Cerritos, CA 90701. Subscription rates: U.S. $24.95, Canada $29.00, all other countries $35.00 (surface mail). Make checks payable in U.S. funds drawn on a U.S. bank. Opinions expressed in by-lined articles do not necessarily reflect the opinion of this magazine or the publisher. Mention of products by trade name in editorial material or advertisements contained herein in no way constitutes endorsement of the product or products by this magazine or the publisher. Circulation Department, (213) 926-9540.

POSTMASTER: Please send change of address form 3579 and undelivered copies to COMPUTER USER Magazine, 16704 Marquardt Ave., Cerritos, CA 90701. Application to mail at second class postage rates is pending at Artesia, CA and additional mailing offices.

PERSONAL/HOME



Jumping on the Bandwagon... With Caution

From preschool to college, computers are becoming *the* educational tool of the decade. Through computer implementation, teachers are able to give students more individualized attention, increase productivity, and introduce the powers of computerization to their computer-shy students. All these accomplishments deserve a pat on the back; however, one thought comes to mind: moderation. Can too much emphasis on computers in education have an ill effect? The answer varies.

After calculators became a very affordable item, every high school and college student owned one. Some math and science professors allowed the use of calculators in figuring complex math problems. They based their decision on the assumption that the student must know the theory to reach the correct answer. The calculator posed no threat—it simply sped up the process. Other professors, however, felt the mental excercise of adding, subtracting, multiplying, and dividing kept the mind sharp, and that the practice of these skills was as equally important as theory. These professors felt students could easily become dependent on calculators for simple functions.

Now that computers are rapidly replacing calculators in the classroom, the same thoughts apply. If students and teachers use computers as tools to enhance the educational process (much like flash cards did 20 years ago), the effect will be positive. On the other hand, if computers interrupt the logical flow of step-by-step problem solving experienced by every student, the results can be devastating. A computer *cannot* replace the human brain; it can strengthen it, educate it, and entertain it, but it cannot and should not replace it.

Although this situation might echo the inklings of a sci-fi thriller, it's a very serious, feasible predicament. What can happen with calculators can happen with computers. After all, computers are faster, have more applications, and can even help you write term papers and reports. For example, there are a lot of programs available that correct spelling, grammar, punctuation... the works. Granted, students have to know what to write about, but does theory outweigh practical reading, writing, and arithmetic skills? The answer is an unequivocal *no*.

A computer program that corrects a student's spelling mistakes doesn't *teach* him or her anything, nor does it narrow the chances of misspelling the same word again. If the student was forced to examine the word—syllabify it, visualize it, write it out—the correction would be etched deeper in the memory and thus reduce chances of misspelling the word again.

Since computers are being introduced at the preschool level, the next generation will literally be raised by these machines—much the way the present generation was raised by the television set. Whether the effect will create an intelligent generation or a computer-dependent one depends on the method by which computers are implemented.

If the situation sounds over-inflated, think about this: How many children can't tell time because of digital clocks?

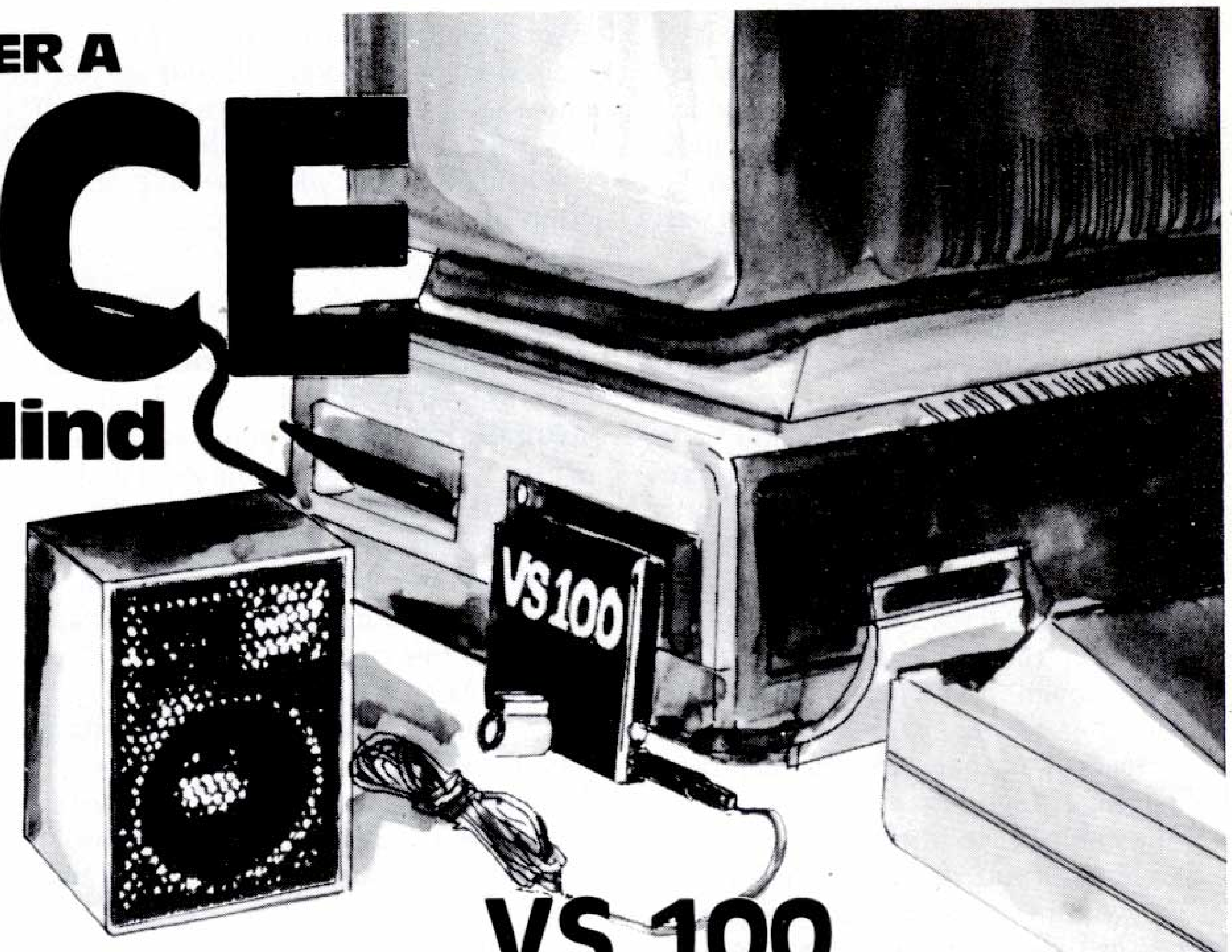
Catherine Semar
Managing Editor

GIVE YOUR COMPUTER A

VOICE

As Well as a Mind

VS-100
VOICE SYNTHESIZER
FOR MODEL I, III, IV



Introducing the "No Compromise" Voice Synthesizer at a price that speaks for itself! Price includes synthesizer module, power supply, 40 page manual, editing and demo software.

Call our NEW Demo Hot-Line
212/296-0399
And hear it for yourself!

VS 100

ALL THIS FOR ONLY! $69.95

Speaker Module Not Included. Add $5.95. Talker 1.4 Text to Speech Software $19.95. Specify tape or disk.

- ☑ Unlimited vocabulary.
- ☑ Automatic inflection.
- ☑ Proven VOTRAX technology.
- ☑ We use the famous VOTRAX SC01 phoneme synthesizer. 4 programable pitch level.
- ☑ Built in audio amplifier with volume control.
- ☑ You can add voice to any basic program in minutes.
- ☑ Super efficient: one single line in Basic will produce a full sentence!
- ☑ Works with any speaker (or add $5.95 for handsome speaker module).
- ☑ Price breakthrough: same performance as units costing hundreds of dollars.
- ☑ Voice editor will help you create unlimited number of words, sound effects, etc.
- ☑ Hundreds of applications—now cost effective in education, robotics, speech therapy, monitoring, games, aid to handicapped, security, prompting...
- ☑ Fully assembled and tested.
- ☑ Ready to plug in and talk.
- ☑ Plugs into 50 pin I/O bus on Model 3
- ☑ As usual our 15-day money back guarantee protects you

NEW!

SPEAK ENGLISH AUTOMATICALLY

TALKER 1.4 TEXT TO SPEECH CONVERTER.
Unlimited vocabulary including: words, letters, symbols and numbers up to 99,999,999,999,999!
Talking is as simple as: PRINT*"I Talk!" Machine Language DISK or TAPE: only $19.95

SUPER CONTEST Win $1000 Royalties for the best talking game. Win $1000 for the best "serious application" software
Clubs: Contact us at (212) 296-5916 if you would like to arrange for a product presentation.

ALPHA Products
79-04 Jamaica Ave., Woodhaven, NY 11421 C (212) 296-5916.

Toll Free Order Line
800-221-0916
Orders Only. NY & info call (212) 296-5916.
Hours: 9-5 Eastern Standard Time

Add $3.00 Per Order For Shipping And Handling. We Accept Visa. Mastercard, Checks, M.O. C.O.D. Add Sales Tax. Overseas, FPO APO; Add 10%. Dealer Discounts Available COD: AD $3.00

CIRCLE NO. 2 ON INQUIRY CARD

More on premiere issue

I'm not sure what type of feedback is of value to you, but allow me to relate to you my introduction to your magazine.

I was 1,500 miles from home attending a funeral. I had extended my trip to assist with the legwork necessary to settle a modest estate. I had run out of clean shirts, underwear, and reading material. I didn't want to purchase a copy of any of the computer magazines I subscribe to. I found myself scouring the racks in the Walden Bookstore in the Ohio Valley Mall in St. Clairsville, OH. It wasn't the "Premiere Issue," the modest $2.95 price, or even the name of the magazine that caught my eye. It was the "For the Tandy/Radio Shack System" that made me pick up the first issue of *Computer User*. I bought it and began reading the departments while waiting for other members of our party to complete their shopping. I found immediate pluses. Same publisher as *Interface Age*. Excellent format. A very promising first "Viewpoint." I enjoyed the departments. I finished all the articles of interest to me in the next three days.

I have a very positive reaction to your first endeavor—so positive, in fact, that my subscription is in the mail. I'll never be interested in *all* the articles in any computer magazine, but I was surprised at how many articles were useful or of interest to me in your premiere issue.

There were two other effects your first issue had on me. First, the article "Power and Versatility Highlight the Model 4" solidified my notion that I must upgrade to that machine, and "Three TRS-80 Workalike Computers Compared" finalized my decision not to purchase a "workalike." Second, I've been shopping for a machine to initiate the kids with. My aim is "computer literacy" and educational computing. Price was a large consideration. "Help Stamp Out Computer Illiteracy......Buy a Personal Computer" and "The MC-10: a Feature-Packed, Mini-Sized CoCo" convinced me the MC-10 would be a good starter. (With an upcoming sale, I have since decided to begin with a 16 KB CoCo for the kids.)

I'm impressed with your first issue. Thanks for the inital effort. I now expect great things from you!

Robert J. Stewart, Jr.
Minot AFB, ND

Addendum

The table printed on page 94 of your premiere issue presented a keyboard memory map matrix for the Models I, III, and 4. You may add six more entries to the table. On the row for memory location 14400, the following keys can be added from left to right, beginning with the column headed "4": BREAK, UP ARROW, DOWN ARROW, LEFT ARROW, RIGHT ARROW, SPACE.

I hope your readers will find use for these additional keys. I have had a lot of fun with them, especially the arrow keys.

Good luck with your new magazine.

Tom Baxter
Burke, VA

Comments on Cookbook series

As a TRS-80 Model III owner, I am naturally interested in publications in which most or all of the material and advertising refer to the unit I use.

I picked up your magazine at a newsstand and was delighted with the content. Your authors are easy to understand, which for this novice, was welcome indeed. As an example, I have a two-drive system and could never understand how the disks function. Now I know. William Allen, author of "The Novice's Disk Repair Cookbook," should write some of the documentation received with some new programs. I am sure they would be much easier to use.

May I suggest at least one or two articles each month be written for the novice in just the manner Mr. Allen wrote "The Novice's Disk Repair Cookbook" because there will always be novices.

Clearance G. Searles
San Francisco, CA

Congratulations on your first issues of *Computer User*. I recently purchased a Model 4 and I hope your magazine will help me get the true power out of this wonderful, yet so poorly documented machine.

I read the plan of action written by Ms. Bruhn in December's issue. I think it is an aggressive one and I already "dream" of the articles promised.

I did notice, however, that there were no provisions made in that plan for any hardware development. I am thinking here about projects like domestic control (security, appliance, etc.), or others that can interest beginners and experts. I hope such articles are not excluded from the magazine's overall plan.

And finally, I would like to address some criticisms to William Allen about "The Novice's Disk Repair Cookbook" (Dec. 83). Figures 2 and 3 on pages 32 and 34 are really confusing. In fact, the two lower dumps of each figure are labeled "Model III TRSDOS 1.3" and yet they are different! The accompanying text does not help differentiating them.

Once again, bravo. But please don't overlook the hardware part of the computer hobby.

Remi Habak
Montreal, Quebec

Thank you for bringing the labeling on the four screen dumps to our attention. The correct labeling on figure 2 should be Model III MultiDOS 1.6, Model I TRSDOS 2.3, Model III TRSDOS 1.3. Figure 3 should be Model III MultiDOS 1.6 GAT, Model III TRSDOS 1.3 GAT, Model I TRSDOS 2.3 GAT.

We hope this error caused no undue inconvenience.—ED

An additional word on Pascal 80

I was pleased to see a generally factual review of Pascal 80 in the December issue of *Computer User*. However, the article contained a misleading warning, indicating that you cannot use it on a single drive system. That is not so. While the author experienced a failure loading to Drive 0, more than half of our users have only one disk drive, and the failure rate of the transfer program is very small. (If users report transfer problems, we send a new disk.)

The review mentions inadequate information on our file handling in our manual. We have since added a tutorial on file handling in response to many user requests; as Louis Buscaglia reports, we are very concerned with supporting Pascal 80 well.

Some of the problems with standard Pascal in general mentioned in the review have solutions in Pascal 80. A fixed-length string can avoid padding on output with blanks if a character 0 is stored after the last legitimate character. Dynamic strings can be created with pointer variables, but they do require 3 bytes per character. The lack of READ...DATA is a nuisance, but there are many ways to program around it that are almost as concise. For example, if S is a string:

Basic
```
10 READ S$
20 DATA "The Data"
```

Pascal
```
S : = 'The Data';
```

or

Basic
```
10 READ$
20 PRINT$
30 DATA "The Data"
```

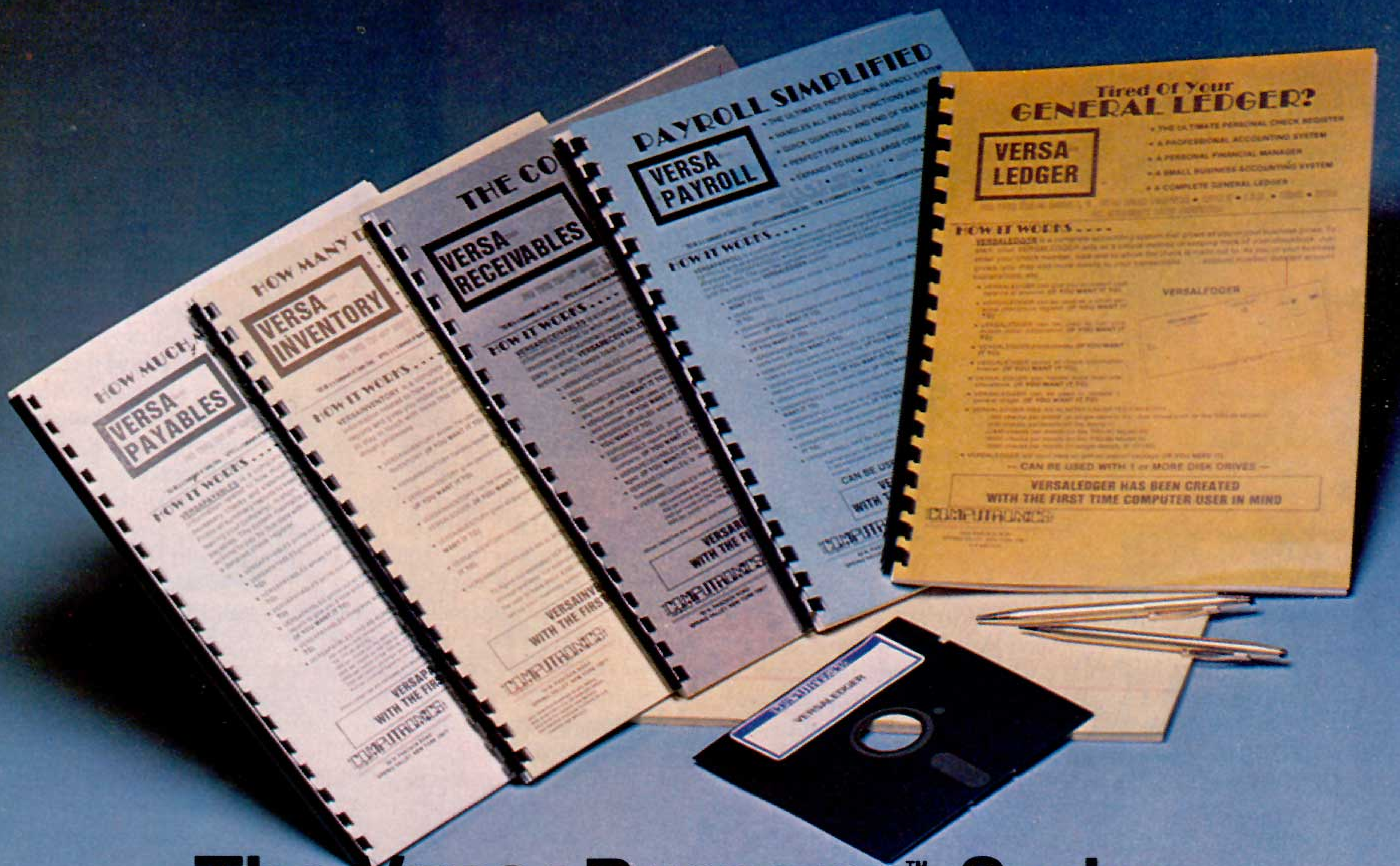
Pascal
```
PROC R (S : string);
BEGIN
  WRITELN(S)
END;
..
R('The Data')
```

A CP/M version of the language is also currently available.

George Blank, President
New Classics Software
Denville, NJ

Introducing the Most Powerful Business Software Ever!

TRS-80™ (Model I, II, III, or 16) • APPLE™ • IBM™ • OSBORNE™ • CP/M™ • KAYPRO™



The VERSABUSINESS™ Series

Each VERSABUSINESS module can be purchased and used independently, or can be linked in any combination to form a complete, coordinated business system.

VERSARECEIVABLES™ $99.95

VERSARECEIVABLES™ is a complete menu-driven accounts receivable, invoicing, and monthly statement-generating system. It keeps track of all information related to who owes you or your company money, and can provide automatic billing for past due accounts. VERSARECEIVABLES™ prints all necessary statements, invoices, and summary reports and can be linked with VERSALEDGER II™ and VERSAINVENTORY™.

VERSAPAYABLES™ $99.95

VERSAPAYABLES™ is designed to keep track of current and aged payables, keeping you in touch with all information regarding how much money your company owes, and to whom. VERSAPAYABLES™ maintains a complete record on each vendor, prints checks, check registers, vouchers, transaction reports, aged payables reports, vendor reports, and more. With VERSAPAYABLES™, you can even let your computer automatically select which vouchers are to be paid.

VERSAPAYROLL™ $99.95

VERSAPAYROLL™ is a powerful and sophisticated, but easy to use payroll system that keeps track of all government-required payroll information. Complete employee records are maintained, and all necessary payroll calculations are performed automatically, with totals displayed on screen for operator approval. A payroll can be run totally, automatically, or the operator can intervene to prevent a check from being printed, or to alter information on it. If desired, totals may be posted to the VERSALEDGER II™ system.

VERSAINVENTORY™ $99.95

VERSAINVENTORY™ is a complete inventory control system that gives you instant access to data on any item. VERSAINVENTORY™ keeps track of all information related to what items are in stock, out of stock, on backorder, etc., stores sales and pricing data, alerts you when an item falls below a preset reorder point, and allows you to enter and print invoices directly or to link with the VERSARECEIVABLES™ system. VERSAINVENTORY™ prints all needed inventory listings, reports of items below reorder point, inventory value reports, period and year-to-date sales reports, price lists, inventory checklists, etc.

VERSALEDGER II™ $149.95

VERSALEDGER II™ is a complete accounting system that grows as your business grows. VERSALEDGER II™ can be used as a simple personal checkbook register, expanded to a small business bookkeeping system or developed into a large corporate general ledger system **without any additional software.**

- VERSALEDGER II™ gives you almost unlimited storage capacity (300 to 10,000 entries per month, depending on the system),
- stores all check and general ledger information forever,
- prints tractor-feed checks,
- handles multiple checkbooks and general ledgers,
- prints 17 customized accounting reports including check registers, balance sheets, income statements, transaction reports, account listings, etc.

VERSALEDGER II™ comes with a professionally-written 160 page manual designed for first-time users. The VERSALEDGER II™ manual will help you become quickly familiar with VERSALEDGER II™, using complete sample data files supplied on diskette and more than 50 pages of sample printouts.

SATISFACTION GUARANTEED!

Every **VERSABUSINESS™** module is guaranteed to outperform all other competitive systems, and at a fraction of their cost. If you are not satisfied with any **VERSABUSINESS™** module, you may return it within 30 days for a refund. Manuals for any **VERSABUSINESS™** module may be purchased for $25 each, credited toward a later purchase of that module.
All CP/M-based Computers must be equipped with Microsoft BASIC (MBASIC or BASIC-80)

To Order:
Write or call Toll-free (800) 431-2818
(N.Y.S. residents call 914-425-1535)

* add $3 for shipping in UPS areas
* add $4 for C.O.D. or non-UPS areas
* add $5 to CANADA or MEXICO
* add proper postage elsewhere

VISA · AMERICAN EXPRESS

DEALER INQUIRIES WELCOME
All prices and specifications subject to change / Delivery subject to availability.

COMPUTRONICS INC.

50 N. PASCACK ROAD, SPRING VALLEY, N.Y. 10977

TRS-80 trademark Tandy Corp. · APPLE trademark Apple Corp. · IBM PC trademark IBM Corp. · OSBORNE trademark Osborne Corp. · XEROX trademark Xerox Corp. · KAYPRO trademark Non-Linear Systems, Inc. · TELEVIDEO trademark Televideo Systems, Inc. · SANYO trademark Sanyo Corp. · NEC trademark NEC Corp. · DEC trademark Digital Equipment Corp. · ZENITH trademark Zenith Corp. · TI PROFESSIONAL COMPUTER trademark Texas Instruments, Inc. · SUPERBRAIN trademark Intertec Corp. · CP/M trademark Digital Research · EPSTON trademark Epson Corp.

CIRCLE NO. 17 ON INQUIRY CARD

by Gary Bellasalmo and Charles White

This column is dedicated to answering your questions about TRS-80 computers and compatible products. If you have a question or a problem you need help with, send it to:

Editorial Department
Computer User
16704 Marquardt Ave.
Cerritos, CA 90701

We'll answer as many questions as we have space for each month, but time limitations won't allow us to contact readers personally.

So, if you are having a recurring problem with a program, want a recommendation before you buy a new product, have a question about computer terminology, or need advice on a particular application, just drop us a note. We'll try to give you an objective, timely answer.

Q. A friend and I were playing with a simple Basic program:

```
10 PRINT "MY NAME IS JOHN"
20 GOTO 10
```

My friend got the message to run across the screen. How did he do it?

A. In Radio Shack Basic, there are two print functions that can be used. One is the comma (,) and the other is the semi-colon (;). If you put a semicolon at the end of line 10, the computer will print the next item right after the first. You will need to add a blank space after JOHN, though, or you will get MY printed right next to JOHN the next time around. The other option is to print in columns. Putting a comma at the end of line 10 will cause the computer to "tab" to the next column each time it goes through the loop. If you leave line 10 alone, the computer will feed to the next line each time through the loop so you'll get a one-column vertical list.

Q. We have a Model 4 with two disk drives and 128 KB of memory. Apparently, the full 128 KB of memory is not directly accessible or useable. Our owner's manual does not discuss the use of the additional 64 KB option. We received no additional literature with the 64 KB add-on purchase. How do we access this additional 64 KB?

A. Your additional memory may be used for a feature called Memdisk. A Memdisk is an area of the computer's random access memory (RAM) that is set aside to be used like a floppy diskette. The Memdisk may be used on the 64 KB Model 4, but its true use is on the 128 KB Model 4. This Memdisk concept will allow you to FORMAT, BACKUP, and COPY files to and from the computer's memory to disk. This gives you fast access to solid state memory, rather than a floppy media transfer. In other words, the computer can find something much quicker in its own memory than it can when searching for a record on diskette. Of course, the data you place in Memdisk is still volatile because it is still random access memory. If you turn off the computer before making a COPY to diskette, you will lose your data. To use the entire 128 KB of memory, it would only be fair to speculate that new programs and patches for older programs may be coming in the future.

Q. My subscription for *Computer User* was mailed several weeks ago and I am wondering why I cannot use the "PODE" service. Your 800 number operator tells me that my subscription will start in six to eight weeks, but must I wait that long to use "PODE"? I am looking forward with anticipation and excitement to being able to share with your other "PODE" communicators.

A. No, you won't have to wait that long to start using "PODE." Here is a description of "PODE," the equipment you'll need, and the procedure to start the service. "PODE" is a subscription service just like *Computer User*. If you are a charter subscriber to *Computer User*, you will get valid "usership" free until December 1, 1984. If you are not a charter subscriber, the normal subscription fee to "PODE" is $10. This will allow you to access our download section, where you can retrieve the programs from *Computer User* and a few other sources at no additional charge. You may also address the E-Mail section of "PODE," where you can send messages to other users. They will receive their messages the next time they log on. "PODE" also has an area for public messages. This is much like a public bulletin board that you would find at your local supermarket, except that these boards can be read from anywhere in the world. A valid user may post or remove his or her own advertisement or message. All you need in the way of equipment is a computer that can run a terminal program and a modem. There are two types of terminal programs for your computer. One is called a dumb-terminal; the other is called a smart-terminal. The smart-terminal program will allow you the ability to download programs from "PODE"; the dumb-terminal types do not. Whether you subscribe by mail or phone, you must still call the "PODE" system and give your subscription information again. Use option <Y> at the top menu of "PODE." After two or three days (to allow for verification of your mailed or phoned-in information), you will become a verified "PODE" user. To summarize, there are two things you must do: first, mail or phone in your name, address, telephone, and password; second, call "PODE" and give the same information to the computer. The next time you call, you should be a "PODE" user. If you have problems with "PODE," you can always leave a message when you log off, or call 213-926-9544 and ask for a "PODE" operator.

Q. I have a Model 100, and in it is a document file called BOOKS.DO. How do I save this file to and retrieve it from cassette tape? The owner's manual seems to have been written for those who are already familiar with Microsoft Basic.

A. In order to save a single file to cassette tape, all you have to do is press <F3> from the top menu. You will be prompted with the words SAVE TO:. . . . . You then type a filename one to six characters long. Be sure that the first character is one of the letters A-Z. If you are in Basic, you can save the file using SAVE"CAS:filename". Of course, the CAS stands for cassette and you supply the filename. This is all you have to do.

H&E COMPUTRONICS INC.

FOR YOUR TRS-80™ • APPLE™ • IBM PC™ • ATARI™ • OSBORNE™ • XEROX™ • KAYPRO™ • ALL COMMODORE Computers TELEVIDEO™ • ZENITH™ • SANYO™ • NEC™ • DEC™ • TI PROFESSIONAL COMPUTER™ • SUPERBRAIN JR.™ • EPSON™
Any CP/M™ Computer with 8" Disk Drives
CP/M-based Computers must be equipped with Microsoft BASIC (MBASIC or BASIC-80)

NEW

BUSINESS PAC 100

100 Ready-To-Run Business Programs

★ All orders processed within 24-Hours
★ 30-Day money back guarantee

(ON CASSETTE OR DISKETTE).....Includes 128 Page Users Manual.....
Inventory Control.....Payroll.....Bookkeeping System.....Stock Calculations.....
Checkbook Maintenance.....Accounts Receivable.....Accounts Payable.....

BUSINESS PAC 100 PROGRAM LIST

	NAME	DESCRIPTION
1	RULE78	Interest Apportionment by Rule of the 78's
2	ANNU1	Annuity computation program
3	DATE	Time between dates
4	DAYYEAR	Day of year a particular date falls on
5	LEASEINT	Interest rate on lease
6	BREAKEVN	Breakeven analysis
7	DEPRSL	Straightline depreciation
8	DEPRSY	Sum of the digits depreciation
9	DEPRDB	Declining balance depreciation
10	DEPRDDB	Double declining balance depreciation
11	TAXDEP	Cash flow vs. depreciation tables
12	CHECK2	Prints NEBS checks along with daily register
13	CHECKBK1	Checkbook maintenance program
14	MORTGAGE/A	Mortgage amortization table
15	MULTMON	Computes time needed for money to double, triple, etc.
16	SALVAGE	Determines salvage value of an investment
17	RRVARIN	Rate of return on investment with variable inflows
18	RRCONST	Rate of return on investment with constant inflows
19	EFFECT	Effective interest rate of a loan
20	FVAL	Future value of an investment (compound interest)
21	PVAL	Present value of a future amount
22	LOANPAY	Amount of payment on a loan
23	REGWITH	Equal withdrawals from investment to leave 0 over
24	SIMPDISK	Simple discount analysis
25	DATEVAL	Equivalent & nonequivalent dated values for oblig.
26	ANNUDEF	Present value of deferred annuities
27	MARKUP	% Markup analysis for items
28	SINKFUND	Sinking fund amortization program
29	BONDVAL	Value of a bond
30	DEPLETE	Depletion analysis
31	BLACKSH	Black Scholes options analysis
32	STOCVAL1	Expected return on stock via discounts dividends
33	WARVAL	Value of a warrant
34	BONDVAL2	Value of a bond
35	EPSEST	Estimate of future earnings per share for company
36	BETAALPH	Computes alpha and beta variables for stock
37	SHARPE1	Portfolio selection model-i.e. what stocks to hold
38	OPTWRITE	Option writing computations
39	RTVAL	Value of a right
40	EXPVAL	Expected value analysis
41	BAYES	Bayesian decisions
42	VALPRINF	Value of perfect information
43	VALADINF	Value of additional information
44	UTILITY	Derives utility function
45	SIMPLEX	Linear programming solution by simplex method
46	TRANS	Transportation method for linear programming
47	EOQ	Economic order quantity inventory model
48	QUEUE1	Single server queueing (waiting line) model
49	CVP	Cost-volume-profit analysis
50	CONDPROF	Conditional profit tables
51	OPTLOSS	Opportunity loss tables
52	FQUOQ	Fixed quantity economic order quantity model
53	FQEOWSH	As above but with shortages permitted
54	FQEOQPB	As above but with quantity price breaks
55	QUEUECB	Cost-benefit waiting line analysis
56	NCFANAL	Net cash-flow analysis for simple investment
57	PROFIND	Profitability index of a project
58	CAP1	Cap. Asset Pr. Model analysis of project
59	WACC	Weighted average cost of capital
60	COMPBAL	True rate on loan with compensating bal. required
61	DISCBAL	True rate on discounted loan
62	MERGANAL	Merger analysis computations
63	FINRAT	Financial ratios for a firm
64	NPV	Net present value of project
65	PRINDLAS	Laspeyres price index
66	PRINDPA	Paasche price index
67	SEASIND	Constructs seasonal quantity indices for company
68	TIMETR	Time series analysis linear trend
69	TIMEMOV	Time series analysis moving average trend
70	FUPRINF	Future price estimation with inflation
71	MAILPAC	Mailing list system
72	LETWRT	Letter writing system-links with MAILPAC
73	SORT3	Sorts list of names
74	LABEL1	Shipping label maker
75	LABEL2	Name label maker
76	BUSBUD	DOME business bookkeeping system
77	TIMECLCK	Computes weeks total hours from timeclock info.
78	ACCTPAY	In memory accounts payable system-storage permitted
79	INVOICE	Generate invoice on screen and print on printer
80	INVENT2	In memory inventory control system
81	TELDIR	Computerized telephone directory
82	TIMUSAN	Time use analysis
83	ASSIGN	Use of assignment algorithm for optimal job assign.
84	ACCTREC	In memory accounts receivable system-storage ok
85	TERMSPAY	Compares 3 methods of repayment of loans
86	PAYNET	Computes gross pay required for given net
87	SELLPR	Computes selling price for given after tax amount
88	ARBCOMP	Arbitrage computations
89	DEPRSF	Sinking fund depreciation
90	UPSZONE	Finds UPS zones from zip code
91	ENVELOPE	Types envelope including return address
92	AUTOEXP	Automobile expense analysis
93	INSFILE	Insurance policy file
94	PAYROLL2	In memory payroll system
95	DILANAL	Dilution analysis
96	LOANAFFD	Loan amount a borrower can afford
97	RENTPRCH	Purchase price for rental property
98	SALELEAS	Sale-leaseback analysis
99	RRCONVBD	Investor's rate of return on convertable bond
100	PORTVAL9	Stock market portfolio storage-valuation program

☐ TRS-80 Cassette Version $99.95
☐ TRS-80 (Mod-I or III), Commodore, Apple, or Atari Versions $99.95
☐ TRS-80 Mod-II, IBM, Osborne and CP/M Versions $149.95

ADD $3.00 FOR SHIPPING IN UPS AREAS
ADD $4.00 FOR C.O.D. OR NON-UPS AREAS
ADD $5.00 TO CANADA AND MEXICO
ADD PROPER POSTAGE OUTSIDE OF U.S., CANADA AND MEXICO

NEW TOLL-FREE ORDER LINE
(OUTSIDE OF N.Y. STATE)
(800) 431-2818

AMERICAN EXPRESS · VISA · master charge

24 HOUR ORDER LINE
(914) 425-1535

H&E COMPUTRONICS INC.
MATHEMATICAL APPLICATIONS SERVICE™

50 N. PASCACK ROAD
SPRING VALLEY, NEW YORK 10977

ASK FOR OUR 64-PAGE CATALOG
DEALER INQUIRIES INVITED

30-DAY MONEY BACK GUARANTEE
ALL PRICES & SPECIFICATIONS SUBJECT TO CHANGE
DELIVERY SUBJECT TO AVAILABILITY

CIRCLE NO. 18 ON INQUIRY CARD

Europe's largest microcomputer show sets '84 date

(Berkeley, CA) Dianne Brock, U.S. Show Coordinator for Europe's oldest and largest microcomputer show, has announced the date for the ninth Micro Expo. The '84 show will take place May 22-26 at the Palais Des Congres (C.I.P.) in Porte Maillot, Paris, France. Commenting on the rapid growth of the show, Brock said, "At the end of the '83 show, companies had reservations for over 90 percent of the available booths for 1984. However, we made special arrangements to double floor space for '84, and we expect another sell-out crowd. Europeans are buying microcomputers at an astonishing rate, and our exhibitors have been very pleased with the business and visibility they have generated in the European market from attending this show." Primarily a business show, Micro Expo draws more than a third of its attendance from the management ranks of European companies. Also visiting the show in great numbers are engineers, educators, and professional people from various disciplines. Approximately 35,000 people are expected to attend the 1984 show. A small sampling of exhibitors already reserved for the '84 show includes Apple Computers, Texas Instruments, Kaypro, Xerox, and Tandy/Radio Shack.

Teachers' micro seminar

(Reston, VA) The National Council of Teachers of Mathematics (NCTM) is conducting special microcomputer seminars during the 1983-84 school year. Each is an intensive two-day program designed to introduce microcomputers to teachers and supervisors of mathematics education at the elementary, intermediate, and secondary school levels. According to NCTM, "All mathematics teachers should acquire computer literacy in order to deal with the impact of computers on their own lives and to keep pace with the inevitable sophistication their students will achieve." The speakers for the NCTM microcomputer seminars were selected for their special interests in teaching with microcomputers. Each has a unique background of practical experience with microcomputers. Together they have prepared a handbook of instructional materials for distribution to seminar participants.

Four new investment services

(Bryn Mawr, PA) NewsNet, the world's largest electronic data base of business newsletters, has added four new services to its "Investment" category. All four services are produced by Idea Publishing of Yardley, PA. Penny Stock Preview includes names and phone numbers of both stock issuing corporations and underwriters through whom penny stocks can be purchased without commissions. It is a monthly publication, but its electronic version is updated and released weekly on NewsNet. Low-priced Stock Alert is the new electronic version of the monthly newsletter, Low-priced Stock Digest. It is available daily on NewsNet as a compilation of stock purchase tips and advice from hundreds of investment newsletters in the field of stock shares priced from 1¢ to $20. Stock Advisors' Alert is a new electronic-only report created expressly for electronic distribution on NewsNet. Its 250 daily issues throughout the year provide compiled stock tips and advice in short, readable form for stock shares priced at more than $20. Market Concensus Alert is another new electronic-only publication. It provides expert commentary on the market in 50 weekly issues, including both short-term analyses and long-term projections.

Panasonic enters peripheral market

(Secaucus, NJ) Panasonic Industrial Company has entered the professional computer peripheral market with the introduction of full lines of printers, plotters, video displays, and data entry terminals. The products will be distributed through an initial network of dealers and representatives that are already active in all 50 states. Panasonic's initial line of products consists of two plotters, two printers, five monitors, and two data entry terminals. Small businesses, ranging from one-room offices to corporations with 500 employees, are the targeted market segment.

Free access to software library

(North Palm Beach, FL) Searchmart Corporation, a South Florida firm specializing in data base development and information retrieval systems, will offer a free access software library that lists, describes, and demonstrates tens of thousands of individual applications and systems software packages on-line. Searchmart's concept is a simple one—an on-line library of systems and applications software updated daily and categorized by manfuacturer, publisher or vendor, operating systems compatibility, protocol requirements, program classification, features, price, and ordering information. Searchmart's free access library allows anyone with data communications capability to search the software data base. The telephone numbers to Searchmart's computers will be publicized in a direct mail and computer magazine advertising campaign. Software manufacturers and vendors will have the opportunity to describe their products and companies on "pages," each page being a 40-character-by-20-line CRT screen.

Floppy copy robot

(Culver City, CA) The brisk demand for microcomputer software has floppy disk duplicating services turning to robots. The robots do not compete with people for jobs; they help people do their own jobs better, and they even create new jobs. Since Ashton-Tate Software Company recently signed on two floppy copy robots, the firm has had to hire three additional people for its copy department. Since the robot started, productivity is up 1,400 percent, customer response time is down from weeks to overnight, and all the required formats for 35 different microcomputers are available in a single machine. An explosive business climate and the industry's proliferating number of disk formats recently moved the firm to investigate ways to boost copy throughput, reduce turnaround time, and consolidate the disk formats of 35 different microcomputers. During an intensive search, Ashton-Tate evaluated numerous robotic disk-copy systems, and selected two systems from Applied Data Communications Inc. of Tustin, CA. Both systems handle 8" and 5-1/4" disks, and they can duplicate object programs and data bases for more than 270 different disk formats. The copy systems, each with a Robotic Disk Handler, cost the firm $30,000 apiece, but each system has returned its entire cost of investment during the first week of operation.

Defense industry on-line

(Bryn Mawr, PA) Companies following Defense Department contracting and R & D programs can now do so with NesNet's electronic information data base. NewsNet has added two publications—Defense Industry Report and Defense R & D Update—both of which can be searched by keywords and read in full text by anyone with a microcomputer, terminal or word processor, connected to telephone lines.

That's what you get with the LNW80 Model 2—undoubtedly the most versatile, powerful and fully equipped microcomputer in its class today. A machine so superior in concept and design, that it will define the standards of microcomputer performance for years to come.

VERSATILITY

The LNW80 2 performs wonders with the most complete library of software available to any microcomputer on the market today. Every LNW80 2 comes complete with this outstanding library of Business Software. **LNW SMALL BUSINESS AND PROFESSIONAL ACCOUNTING SERIES™** General Ledger, Accounts Receivable, Accounts Payable, Payroll; **ELECTRIC SPREADSHEET™; ELECTRIC PENCIL™ WORDPROCESSOR; MICROTERM™ MODEM PROGRAM; CHART EX™ HIGH RESOLUTION BUSINESS GRAPHICS CHARTING PROGRAM; CP/M 2.2®; DOSPLUS™; LNW-BASIC™; MICROSOFT BASIC.™** In addition to a comprehensive line of LNW80 2 Software, it is also fully compatible with software from TRS80® (Models 1, 3, 4), CP/M® and Cromemco® worlds—a capability which gives you access to the most extensive and mature libraries of business, scientific, engineering and entertainment software applications. So no matter how far you expand into user applications, the LNW80 2 will expand right along with you.

POWER

The LNW80 2 performs miracles with the computing power of 96K RAM (standard) of user memory matched with a mass storage capability which handles 5¼" floppy disks and 5¼" hard disk drives. And while the unit comes with built-in controllers for 5¼" and 8" floppy disks (single/double sided, single/double density, up to 4.5 Megabytes capacity), the LNW80 2 also gives you the unique ability to read and write diskettes from a greater variety of other popular computers than does any other microcomputer. So regardless of how big you grow, you will never end up with thumb-twiddling down time while you expand to a more powerful system. The LNW80 2 will always have enough muscle to handle your biggest and toughest jobs.

FULLY EQUIPPED

The LNW80 2 was developed to anticipate the needs of both expansion and compatibility. So the computer was designed with enough built-in features to keep you from having to spend a small fortune as you move down the road to higher levels of user sophistication. Standard features include high and low resolution graphics in both color and black-and-white, an asynchronous serial communication channel, and a wide variety of tape, printer, monitor and hardware expansion ports. In addition, the LNW80 2 contains an array of quality construction features that fully justify its remarkable one-year limited warranty.

So if you're looking for a microcomputer that will satisfy your performance needs as you grow and develop, take a long, hard look at the LNW80 2. It's the one microcomputer built to meet the challenges of tomorrow—for a long time to come.

For more information and the name of the dealer nearest you, write or telephone:

LNW Computers
2620 Walnut, Tustin, California 92680
Telephone: 714/544-5744

SPECIAL ANNOUNCEMENT

80x24 HARDWARE VIDEO NOW STANDARD. MEMORY EXPANDABLE TO 160K.

$1395. computer keyboard only.

$1995. computer keyboard, 12" monitor and dual 5¼" single-sided, slimline disk drives.

STATE-OF-THE-ART ENGINEERING STATE OF TOMORROW PERFORMANCE.

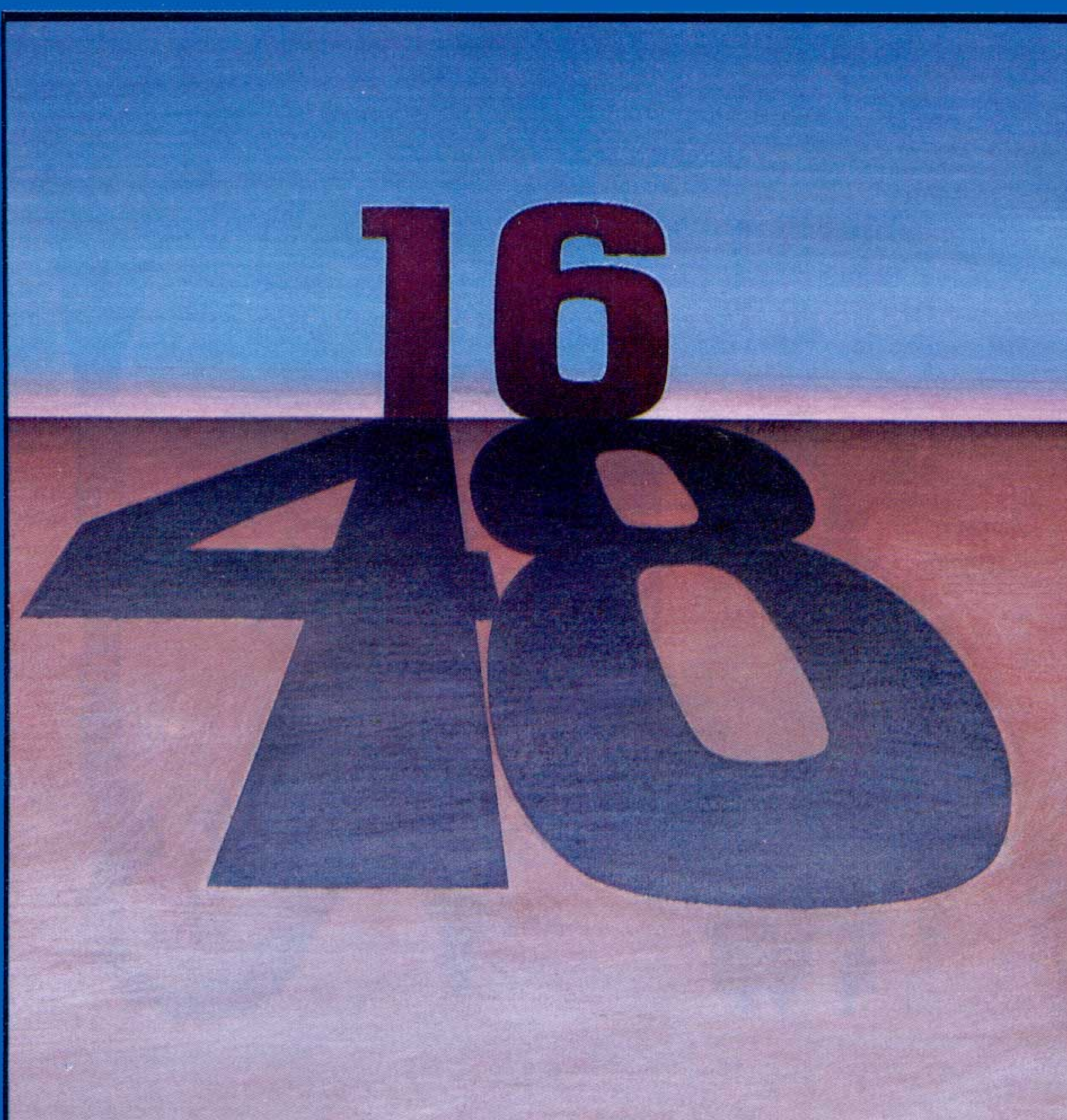
LNW COMPUTERS

CIRCLE NO. 26 ON INQUIRY CARD

BUILDING THE SUPER MICRO 1

If you've outgrown the capabilities of your Model III and don't want to spend thousands of dollars on a Model 4, you have one option: upgrade your Model III. This is the first of five-part series on upgrading.

by Richard Green



If you are one of the many people whose needs have exceeded the capacity of their Model III, there is an option to spending thousands of dollars for a new computer. The Model III can easily be upgraded to have a full 48 KB of memory, more than two million bytes of disk storage, a serial interface, and CP/M capability. The upgraded computer can have all this, and it can run up to twice as fast as an original, unmodified Model III.

With this article, I begin a five-part series that will show how to make all of these upgrades on your Model III. No technical knowledge is required. All you need is a modicum of dexterity and a few hand tools.

The primary value of doing an upgrade yourself is the savings in cost. While an upgrade of this magnitude is not cheap, the cost is a fraction of what a new computer with all these features would cost.

Before you begin the task of upgrading your computer, it will be necessary to purchase 16 4116 RAM chips ($3-$4 each) that are rated to operate at 200 nanoseconds (nS). Chips of this description are available from many different manufacturers including Texas Instruments, Motorola, and NEC, but the specific brand is not important. 4116 chips can be purchased from any electronic supplier.

To install the memory, it is necessary to open the Model III. I should point out that opening the case of the Model III will invalidate the Radio Shack warranty. Also, some Radio Shack service centers refuse to work on any computer that contains any components not installed by an authorized Radio Shack service center. So, if your computer is less than 90 days old, or if you don't have access to a repair center that will work on your modified computer, you might want to reconsider your decision. Otherwise, with just a little patience and care, you can install 32 KB of RAM in a few minutes.

Before you start, make sure that the computer is unplugged from the electric source.

Place the computer on a soft surface on a stable work area. (A piece of carpeting on the kitchen table will work fine.) With the computer faced away from you, a single phillips-head screw is visible on the back of the case, centered near the top. Remove this screw.

Now lay the computer on its back (see figure 1). On the bottom of the computer there are 10 phillips-head screws that

hold the case top to the bottom. Each of these screws is recessed into the case bottom. Notice that the four screws that are in the black rubber feet are *not* to be removed.

As you remove the 10 screws, note the position of each. There are three different size screws that hold the case together. Along the front are three short machine screws. Along each side there is one long machine screw. The remaining five screws are short sheet metal screws of equal size. One of the sheet metal screws is *under* the warranty seal, which is the black label at the left rear corner of the computer. Place the 10 screws into a small plastic bag and tape the bag to the bottom of the case to prevent losing the screws.

The case top and bottom are now free. *Be careful moving the computer—a wrong move could break the display tube.*

Grasp the case at the sides, holding the top and bottom together. Turn the computer onto its feet, with the back toward you. Carefully lift the top straight up while looking through the ventilation slots to ensure that the neck of the picture tube does not hang on any of the internal wiring. Lay the top on its left side (see figure 2).

Remove the video connector and the ground wire if your computer has one (some early Model IIIs do not have a separate ground wire to the video card). The only reason to remove these connectors is to prevent damaging the display tube should the computer be moved during the subsequent operations.

Your computer may or may not have a radio frequency (RF) interference shroud installed. If the back of the computer is a featureless piece of sheet metal, that is the RF shroud. Remove it by taking out the six sheet metal screws that hold it in place.

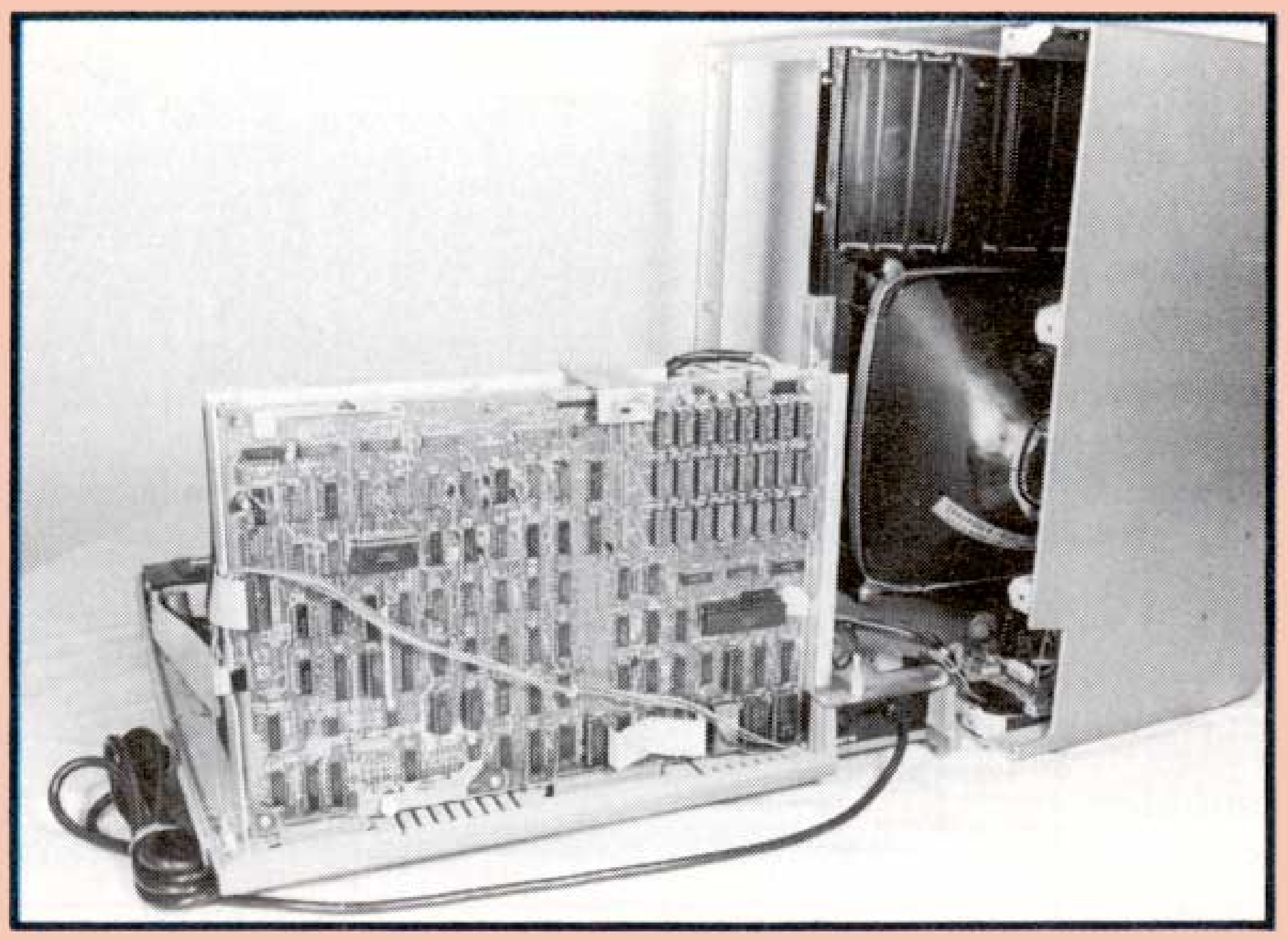


Figure 2.
With the back of the computer toward you, carefully lift the top straight up, and lay it on its left side.

Once the main PC (printed circuit) board is exposed (see figure 3), you are ready to install the 4116 RAM chips. These 16 chips are inserted into the 16 empty sockets at the upper right corner of the main PC board. It makes absolutely no difference in what order the chips are installed. *However*, the chips each have a reference notch at one end. Each chip *must* be installed with the notch at the *top*. If the chip is upside down, it will not work and the chip may be damaged.

The chip is installed by simply pressing it into the socket as far as it will go. To judge whether or not the chip is seated, compare it with one of the chips in the top row. If the chip appears to be seated as far as one of these, it is in far enough.

Before you press the chip into the socket, be sure that all the pins are aligned. If the pins are spread too far apart to slip into the socket, press the pins on one side against a hard surface to bend them until they are perpendicular to the face of the chip. Do this to both sides of the chip, and the pins will align with the socket. If you do happen to bend a pin, just pull the chip from the socket, straighten the pin with a small needle-nose pliers, and reinsert the pin into the socket. The 4116 chips are fairly rugged, so you are unlikely to damage one inadvertently.

Of course, you want to replace the button on the keyboard that now reads 16 K with one that reads 48 K. These are availale only at Radio Shack. They cost the tremendous sum of $1.50. To replace the button, remove the six screws that hold the keyboard trim plate in place. The button is held in place by two tabs that are simply bent over on the underside of the trim plate. Straighten the tabs with your fingers, replace the button, and bend the tabs to secure the new button. Replace the screws to hold the trim plate in place.

That's all there is to the memory upgrade.

Reassemble the computer by laying the top, still on its side, next to the bottom. Carefully plug the video connector and ground wire into the video board. Now lift the top into place, again watching through the vent slots to be sure not to strike the neck of the display tube on anything. With the top in position, turn the computer onto its side and replace the 10 screws in the bottom. Be sure to replace the eleventh screw at the top center of the back because this screw secures the main PC board, preventing it from being jarred.

Plug the computer into a power source and turn it on.

When the prompt appears, type in ? MEM and hit ENTER. The display should show "48085" confirming that you now have a 48 KB Model III.

You may want to test the memory chips for integrity. There is some question whether or not this is necessary. I have upgraded several Model Is and

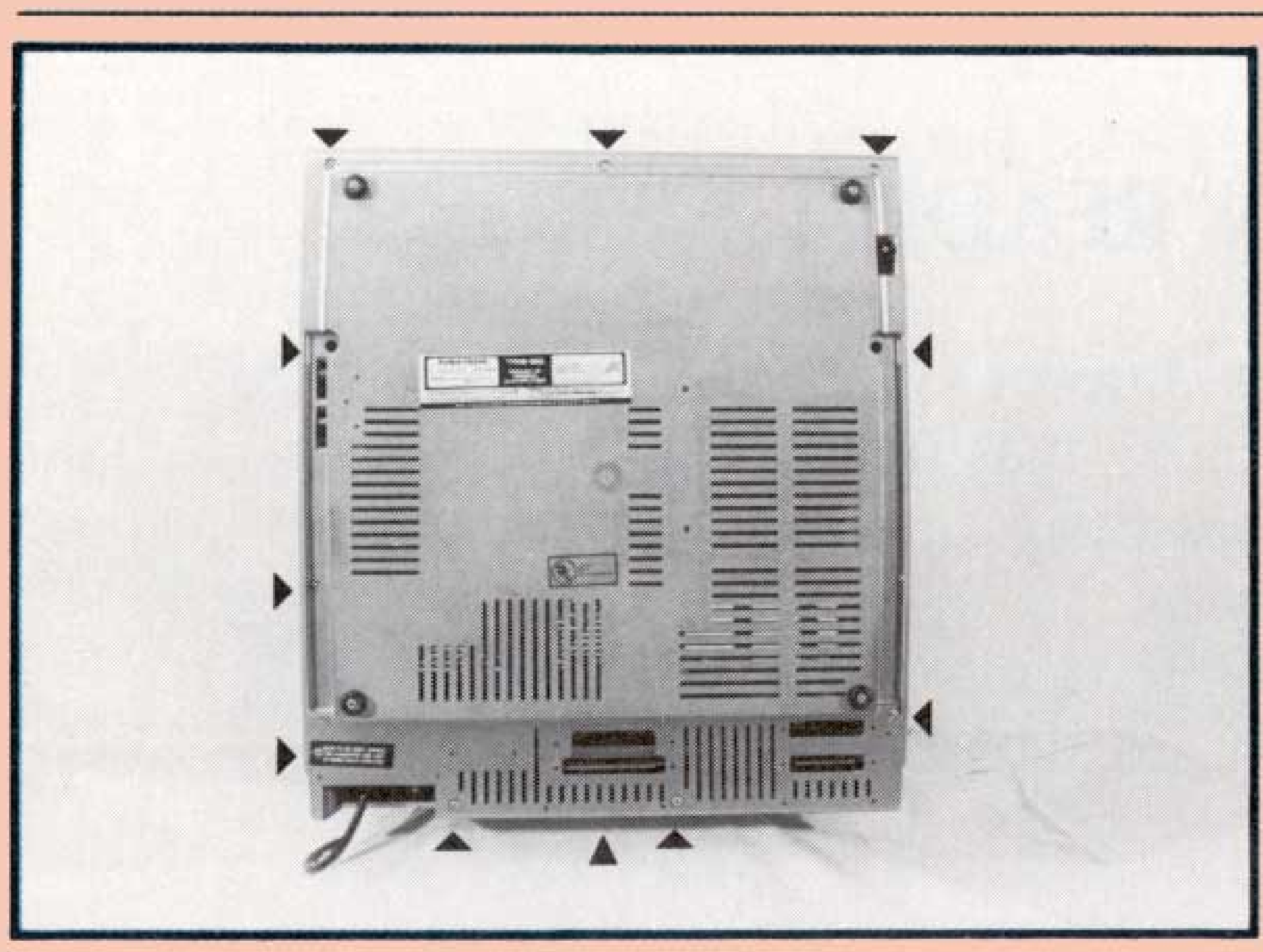


Figure 1.
There are 11 screws to be removed. Ten are in the bottom, and one is centered near the top on the back of the case. Notice that one screw is under the warranty label.

TRS-80 ARTICLES WANTED

The editors of COMPUTER USER are now accepting unsolicited articles and query letters to consider for publication. The primary focus is on applications for TRS-80 computers—unique and unusual methods wherein TRS-80 owners can obtain optimum use of their systems. The fields of education and home use are emphasized. Some business applications will also be included. Four system areas will be covered: Personal/Home (Models I, III, 4), Personal/Business (Models II, 12, 16), Color Computers, and Pockets/Portables. Software, peripherals, and TRS-80-compatible equipment will be included.

Good quality program listings, accompanied by articles explaining the programs, are sure bets for serious consideration during our screening process. The listings should be 60 characters wide, if possible, with no wrap-around lines. Unlined paper and a new ribbon should be used. Sample runs should also be included. In the article, variables should be described. The system utilized in composing the program should be detailed—operating systems, language type and version, and any necessary peripherals. A diskette or cassette version of the program must also be submitted.

Submittals should be prefaced by a brief synopsis of the article. Manuscripts should be typed or printed out double-spaced with one-inch margins. Minimum text length is eight pages, whether or not the article is accompanied by a program listing. Photos should be numbered and have a brief description attached to each. Tables, listings, etc. should be on separate pages and each should have a caption. Authors must submit a statement of background and expertise.

The publisher assumes no responsibility for artwork, photos, or manuscripts. No acknowledgment is made unless the submission is accompanied by a large, stamped return envelope.

The submittals should be addressed to: Editorial Department, COMPUTER USER, 16704 Marquardt Ave., Cerritos, CA 90701.

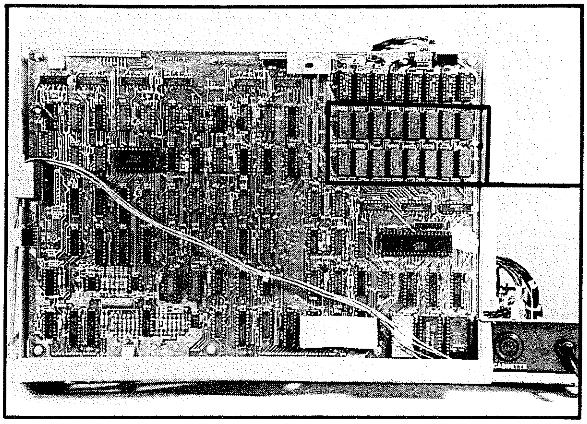


Figure 3. The two rows of sockets where the chips are to be inserted are at the upper right corner of the PC board. They are outlined in the photograph.

```
20  T=&H0000:GOTO 40
30  T=&HFF
40  FOR N= -32768 TO -1
50  POKE N, T
60  S=PEEK(N)
70  IF S<>T GOTO 100
80  NEXT N
90  IF T<>O THEN GOTO 20
    ELSE GOTO 30
100 LPRINT"MEMORY FAULT AT "
    ;N;" SHOULD BE ";T;"WAS ";S
110 GOTO 80
```

Notice that the above program is an infinite loop. The only way to stop the execution of the program is to press the BREAK key.

All that this program does is insert all 1's into each of the memory locations in the new chips. It then looks at the memory location to see if it does, indeed, hold all 1's. The second loop does the same thing except it inserts all 0's. If the memory location does not hold all 1's or all 0's, the location (address) and the value in that location are printed out.

Figure 4. Program listing

Model IIIs and I have never found a bad memory chip. If you would like to exercise the memory, type in the Basic program shown in figure 4. It will take only a few minutes of running time to be sure that all the memory locations have been tested. If any do not work, the memory location and the tested value will print out. Should the test show a bad memory location, you will have to replace the failed chip. The running of the test programs may be stopped at any time by pressing the BREAK key. To "burn-in" the new memory, just let the program run. If you want to test the memory with random insertions, delete line 30 and change lines 20 and 90 to read as follows:

```
20 T = RND (255)
90 GOTO 20
```

Next month I'll discuss installing four half-height disk drives. ☐

Richard Green is a pilot, currently on leave from a major airline. He is presently working as an income tax consultant, and also provides advice to small businesses in the areas of start-up, aviation, and office automation.

ATTENTION READER:

In the December issue of Computer User on page 106 an advertisement was mistakenly run which was time/offer sensitive.

We apologize to our readers and any customers of Montezuma Micro which may have been inconvenienced.

MONTEZUMA MICRO

P.O. Box 2169
Camp Verde (Lizard Flats)
Arizona 86322

WE KEEP YOU RUNNING

COMPUTERS

	Model 4 with 2 drives & RS-232 64K Complete	1495
26-1067	16K Model 4, Pure Radio Shack	849
26-1068	1dr Model 4, Pure Radio Shack	1444
26-1069	2dr Model 4 64K RS232, Pure Radio Shack	1650
26-1080	**NEW!** Model 4 Portable	1529
26-3003	64K Color Computer	340
26-3011	MC-10 Micro Color Computer	99
26-3026	16K Color Computer 2	199
26-3027	16K Extended BASIC Color Computer 2	275
26-3590	Pocket Computer 3	84
26-3601	Pocket Computer 2	169
26-3650	Pocket Computer 4	59
26-3801	Model 100 8K Portable Computer	679
26-3802	Model 100 24K Portable Computer	849
26-4004	Model 12 2-Drive	2599
26-4005	Model 12 2-Drive	3199
26-6004	Model 16B 1-Drive	4249
26-6005	Model 16B 2-Drive	4928
26-6006	Model 16B with built-in 12Mb Hard Disk	5949
26-6050	DT-1 Terminal	599

LNW Model II Computer, 112K, RS-232, TRSDOS Compatible Complete with CP/M 2.2 and lots of FREE Software 1488

ZORBA Portable, 2 DSDD drives, 80x25, 4Mhz, graphics, 7 green phosphor screen, RS-232, IEEE-488 & Centronics ports, with WordStar, MailMerge, SuperSort, CalcStar 1695

MODEMS

26-1084	Model 4P Modem Board	128
26-1173	DC Modem II	169
26-1174	Acoustic Coupler Modem	127
26-1175	DC Modem 1B	85
76-1005	DC-1200 High Speed Modem	595
76-1009	Auto Dial Attachment for DC-1200	127
	Anchor Mark I Modem, 300 Baud	89
	Password 300/1200 Baud Modem	399
	Hayes Smartmodem, 300 Baud	233
	Hayes Smartmodem, 1200 Baud	525

PERIPHERALS

	LNW Model 1 Expansion Interface. 32K & RS-232	329
	12" Green 18MHz NonGlare Video Monitor	87
	12" Amber 20MHz NonGlare Video Monitor	99
	Video cable for Model 1 keyboard-to-monitor	7
16-0230	13" Remote Control Cable Ready Color TV	285
16-0501	Model 10 Video Tape Recorder w/Remote	449
26-1130	Model 1,3,4 5MB Hard Disk Primary	1699
26-1131	Model 1,3,4 5MB Hard Disk Secondary	1529
26-1183	Model 100 Bar Code Reader	85
26-1208	CCR-81 Cassette Tape Recorder	51
26-1410	Model 100 Modem Cable	17
26-3007	Color Computer Carrying Case	17
26-3008	Color Computer Joysticks, Per Pair	21
26-3012	**NEW!** Deluxe Joystick/each	34
26-3022	Color Computer Disk Drive 0 Kit	339
26-3023	Color Computer Disk Drive 1,2 or 3	237
	Same as above except our brand Save $$$$	199
26-3024	Color Computer Multi-Pak Interface	153
26-3025	Color Computer Mouse	43
26-3029	Disk Drive 0 for the Color Computer 2	329
26-3503	PC 1 Cassette Interface	25
26-3508	PC 1 Carrying Case	13
26-3608	PC 2 Carrying Case	25
26-3612	PC 2 RS-232C Interface	169
26-3615	PC 2 4K RAM Module	33
26-3616	PC 2 8K RAM Module	118
26-3651	PC 4 Cassette Interface	33
26-3653	PC 4 1K Ram Module	16
26-3654	PC 4 Carrying Case	6
26-3804	Model 100 AC Adapter	5
26-3805	Model 100 Acoustic Coupler	34
26-3809	Model 100 Carrying Case	42
26-3816	Model 100 8K RAM Module	101
	Same as above except our brand Save $$$$	79
26-4152	Model 2,12,16 12Mb Hard Disk Primary	2545
26-4153	Model 2,12,16 12Mb Hard Disk Secondary	2120
26-4160	Model 2 1-Disk Expansion Bay	674
26-4162	Model 2 3-Disk Expansion Bay	1379
26-4165	Model 12, 16 SlimLine 1-Drive Expansion	1104
26-4166	Model 12, 16 SlimLine 2-Drive Expansion	1783

FURNITURE

26-1305	Model 3 System Desk	152
26-1308	Universal Printer Stand	85
26-1324	Computer Table	60
26-1325	Platform	26
26-4301	Model 2 Desk	299
26-4303	Deluxe System Desk	229
26-4304	System Desk Drawer	111
26-4305	Deluxe Printer Stand	128
26-4306	Terminal Stand	153
74-0550	Docu-Rack Stand	51

PRINTERS & ACCESSORIES

26-1155	Quick Printer II	69
26-1158	Daisywheel II	1525
26-1190	Plotter	424
26-1191	Multi-Pen Plotter	1695
26-1192	GCP-115 Color Graphics Printer	169
26-1193	Flatbed Plotter/Printer	845
26-1195	Digitizer	381
26-1196	Color Computer Graphics Input Tablet	297
26-1250	DWP-410 Daisywheel Printer	1099
26-1251	DMP-400 140cps Printer	715
26-1252	DMP-500 220cps Printer	1099
26-1253	DMP-100 50cps Printer	339
26-1254	DMP-200 120cps Printer	594
26-1255	DMP-120 120cps Dual Mode Printer	424
26-1256	DMP-2100 160cps Dual Mode Printer	1695
26-1257	DWP-210 Daisywheel Printer	679
26-1260	Envelope Feeder for the Daisywheel II	1015
26-1261	TP-10 Thermal Matrix Printer	85
26-1266	CR-510 Computer Card Reader	1440
26-1267	DMP-420 140cps Triple Mode Printer	849
26-1268	CGP-220 Color Ink Jet Printer	594
26-1269	PTC-64 64K Printer Controller(Buffer)	212
	All $29 Printwheels	25
	All $39 Printwheels	33
26-1440	Single Bin Sheet Feeder	845
26-1441	Bi-Directional Tractor	144
26-1443	Bi-Directional Tractor	127
26-1447	Bi-Directional Tractor	208
26-1448	Single Bin Sheet Feeder	999
26-1455	Acoustic Cover	339
26-1459	Bi-Directional Tractor	204
26-1401	Model 1, M3, M4 Printer Cable	29
26-1408	RS-232C Cable	16
26-1409	Model 100 Printer Cable	13
26-4401	Model 2, M12, M16 Printer Cable	29
26-1490	10' RS-232C Cable	25
26-1491	25' RS-232C Cable	33
26-1492	50' RS-232C Cable	46
26-1493	100' RS-232C Cable	76
26-1495	RS-232C Cable Extender	25
26-1496	RS-232C Null Modem Adapter	25
26-1498	Parallel Printer Switch	102
26-1499	RS-232C Selector Switch	128
26-3505	PC 1 Printer	108
26-3591	PC 3 Printer	102
26-3605	PC 2 Printer	187
26-3652	PC 4 Printer	68

C.ITOH

8510	Prowriter 120cps Friction/Tractor Parallel	359
1550	Prowriter 2, 120cps for Wide Paper Parallel	699
F10-40	Starwriter 40cps Daisywheel	1299
F10-55	Printmaster, 55cps Daisywheel, FAST	1499
F10	Bi-Directional Tractor, with Plastic Cover	210
F10	Single Bin Sheet Feeder	888
8600	Near Letter Quality 180cps 2-Color	899
GX-100	Gorilla Banana 50cps	229
CX-4800	Printer Plotter 4-Color	599

ANADEX

DP-9620	180cps Printer	1199
DP-9625A	200cps Triple Mode Printer	1299
WP-6000	285cps Near Letter Quality Printer	2099
DP-6500	500cps Super Trick Includes Tractor	2299
	Tractor Assembly for WP-6000	129
	Automatic Sheet Feeder for WP-6000/6500	925

We have dust covers for most every printer we sell. Be sure to order one with your new printer to keep it like new for years

MONTEZUMA'S REVENGE

This month we offer the Holmes VID-80 with CP/M 2.2 and 112K RAM for the Model III Now you can be the CP/M King and/or Queen on your block and save $125 by installing it yourself Complete instructions included

See the Holmes ads in 80 Micro.

Retail Price $524.50 **YOUR COST ONLY ... $ 399**

SOFTWARE

	NEW CP/M 2.2 by Montezuma Micro For The Model 4	199
	NEW MBASIC for the Model 4	149
26-0310	TRSDOS 2 3 For The Model 1	12
26-0312	TRSDOS 1 3 For The Model 3	12
26-0313	TRSDOS 6 0 For The Model 4	12
26-1507	Stockpack	42
26-1510	Trendex	51
26-1512	**NEW!** Target PlannerCalc	85
26-1515	**NEW!** pfs:file	106
26-1516	**NEW!** pfs:report	85
26-1518	**NEW!** pfs:file for the Model 4	107
26-1520	**NEW!** Model 4 VisiCalc	212
26-1521	VisiCalc Business Forecast	85
26-1530	**NEW!** Model 4 Multiplan	169
26-1540	**NEW!** Model 3 General Ledger	169
26-1541	**NEW!** Model 3 Accounts Receivable	169
26-1543	**NEW!** Model 3 Payroll	169
26-1552	General Ledger	85
26-1553	Inventory Control I	85
26-1554	Accounts Payable	128
26-1555	Accounts Receivable	128
26-1556	Disk Payroll	169
26-1557	Concrete Take-Off	49
26-1558	Business Mailing List	85
26-1559	Manufacturing Inventory Control	169
26-1560	Fixed Assets	68
26-1562	Profile	68
26-1563	Scripsit	85
26-1564	Mailgram	34
26-1565	Microfile	85
26-1566	Model 1 VisiCalc	85
26-1568	Medical Office Systems	254
26-1569	Model 3 VisiCalc, Enhanced Version	169
26-1579	Real Estate	85
26-1580	Project Manager	85
26-1581	Personnel Manager	85
26-1582	Time Manager	85
26-1584	Checkwriter 80	85
26-1585	Business Checkwriter	127
26-1588	Videotex Plus	42
26-1589	MICRO/Courier	127
26-1590	SuperSCRIPSIT	169
26-1591	Scripsit Dictionary	126
26-1592	Profile Plus	169
26-1593	Profile Plus LD/HD Version	254
26-1594	Desktop/Plan-80	169
26-1595	**NEW!** SuperSCRIPSIT For The Model 4	169
26-1596	**NEW!** Scripsit For The Model 4	85
26-1597	Business Graphics Pak	148
26-2010	Mod 3 BASIC Programming Course	25
26-2011	EDAS, Tape Version	25
26-2012	Assembly Language Development Course	126
26-2013	EDAS, Disk Version	29
26-2014	Mod 3 Disk Course	25
26-2017	**NEW!** Assembly Language Course, Tape	34
26-2018	**NEW!** Assembly Language Course, Disk	59
26-2022	**NEW!** Power Tool	43
26-2023	**NEW!** Dot Plot	17
26-2150	Introduction to BASIC	99
26-2200-1	Model 1, Model 3 FORTRAN	85
26-2201	Model 1 FORTRAN	85
26-2203	COBOL	169
26-2204	Compiler BASIC	126
26-2205	PILOT	85
26-2206-7	COBOL Runtime Disk	25
26-2208-9	Compiler BASIC Runtime Disk	17
26-2210	**NEW!** BASCOM	169
26-2211	**PASCAL**	212
26-2212	**NEW!** PASCAL For The Model 4	169
26-2213	Model 1 LDOS	109
26-2214	Model 3 LDOS	109
26-2216	CP/M Plus For The Model 4	126
26-GOOD	The Good CP/M for the Model 4(MM)	199
26-2217	CBASIC For The Model 4 (Requires CP/M)	85
26-2220-23	Videotex	26
26-2224	Videotex Compuserv Kit	16
26-2709	Color Computer PILOT, Tape	51
26-2710	Color Computer PILOT, Disk	68
26-2718	PILOT III	101
26-2721	Color Computor LOGO, Disk	85
26-2722	LOGO ROM Pack	43
26-3019	Diagnostic ROM Pack	17
26-3030	**NEW!** OS-9 For The Color Computer	59
26-3036	**NEW!** BASIC - 09 For The Color Computer	85
26-3821	**NEW!** Model 100 Learning Lab	25

ALL Color Computer GAMES 25% OFF Catalog Price
ALL PC SOFTWARE 30% OFF Catalog Price

26-4501	General Ledger I	135
26-4502	Inventory Management System I	135
26-4503	Payroll	270
26-4504	Accounts Receivable	199
26-4505	Accounts Payable	199
26-4506	Mailing Liat	44
26-4507	Mailing List II	101
26-4508	Medical Office Systems	637
26-4509	Manufacturing Inventory Control	637
26-4510	Versafile	59
26-4511	VisiCalc	254
26-4512	Profile II	152
26-4513	Job Costing	126
26-4514	Order Entry	169
26-4515	Profile II Plus	254
26-4516	Profile Training Guide	58
26-4517	Profile Plus Upgrade	102
26-4520	Time Accounting	424
26-4521	VisiCalc	254
26-4525	Electronic Broker	850
26-4531	SCRIPSIT 2 0	254
26-4532	SCRIPSIT Utility Disk	109
26-4534	SCRIPSIT Dictionary	169
26-4536	SCRIPSIT Plotter Driver	42
26-4540	Statistical Analysis	85
26-4545	Litigation Support	254
26-4550	**NEW!** Business Graphics Anylsis Pak	212
26-4554	Accounts Receivable	107
26-4555	Menu Generator	33
26-4556	Profile Forms	106
26-4557	Profile Archive	42
26-4558	Profile Prosort	126
26-4559	Prolook	169
26-4560	WESTLAW	211
26-4580	MultiPlan, Model 2 & Model 12 Version	212
26-4601	General Ledger	424
26-4602	Inventory Control System	254
26-4603	Payroll	509
26-4604	Accounts Receivable	424
26-4605	Accounts Payable	424
26-4607	Order Entry/ICS	424
26-4608	Sales Analysis	254
26-4621	Personnel Search	254
26-4701	FORTRAN	254
26-4702	EDAS	169
26-4703	COBOL	254
26-4704	COBOL Run-Time Disk	34
26-4705	Compiler BASIC	169
26-4706	Compiler BASIC Run-time Disk	26
26-4707	COBOL Generator	850
26-4708	Report Generator	254
26-4710	Program Editor	67
26-4711	**NEW!** PASCAL MT ♆	361
26-4712	Assembly Language Development System	211
26-4713	EDAS 1	43
26-4714	ReformaTTer	212
26-4716	BiSYNC 3780	850
26-4718	Term Administration	???
26-4721	Videotex For The Model 2, 12, 16	42
26-4725	**NEW!** BASCOM BASIC Compiler	169
26-4730	BiSYNC 3270, Version 2	1270
26-4740	CP/M Plus	211
26-4742	CBASIC	84
26-4802	Inventory for the Hard Disk	339
26-4831	SCRIPSIT for the Hard Disk	339
26-4834	Dictionary for the Hard Disk	169
26-4835	SCRIPSIT for Thinline Drives	339
26-4910	TRSDOS 2 0	21
26-4912	TRSDOS 4 2	21
26-6100	COBOL Development System	254
26-6101	COBOL Run-time Diskette	34
26-6105	TRSDOS 16	34
26-6201	General Ledger	509
26-6203	Payroll	594
26-6204	Accounts Receivable	509
26-6205	Accounts Payable	509
26-6207	Order Entry/ICS	509
26-6208	Sales Analysis	339
26-6209	Job Costing	169
26-6302	Inventory	339
26-6309	Manufacturing ICS	722
26-6401	XENIX Development System	637
26-6455	COBOL	599
26-6457	MBASIC	254
26-6480	MultiPlan, Model 16 & XENIX	296
	Electric Pencil 2 0, M1, M3	69
	Blue Pencil, M1, M3	69
	Red Pencil, M1, M3	69
	HEXSPELL 2	80
	HEXMAN	60
	Modem 80, A Powerful Communications Program	35

CP/M

CP/M 2.2 for the Model 4 The **VERY BEST ONE**	199
MBASIC for the Model 4	149
P & T CP/M 2 2m Floppy, M2-12-16	180
P & T CP/M 2.2m Hard, M2-12-16	225

All **MICROPRO** Software Available. Call For Details

BOOKS

26-2102	Model 1 Level II Manual	5
26-2103	Model 1 Technical Manual	13
26-2109	Model 3 Technical Manual	8
26-2110	Model 4 Technical Manual	21
26-2111	Model 3 DOS Manual	6
26-2112	Model 3 BASIC Instruction Manual	5
26-2114	Sourcebook, New Edition	4
26-2115	Newsletter Book 1980	4
26-2117	Model 4 DOS Manual	15
26-2240	Newsletter Book 1981	8
26-3191	Color Computer Manual	5
26-3192	Extended Color Computer Manual	
26-4920	Model 2 Operators Manual	26
26-4921	Model 2 Technical Reference Manual	26
26-4922	Model 12 Operating Manual	59
26-6040	Model 16 Operators Manual	64
26-6041	Model 16B Operating Manual	72
62-2084	TRS-80 Pocket BASIC Handbook	6
	TRS-80 Disk & Other Mysteries	18
	Microsoft BASIC Decoded	24
	BASIC Faster & Better	24
	BASIC Faster & Better Demonstration Disk	15
	BASIC Faster & Better Library Disk	15
	BASIC Disk I/O Faster & Better	24
	The Custom TRS-80	24
	TRSDOS 2 3 Decoded	24
	Machine Language Disk I/O	24
	How Do It On The TRS-80	24
	TRS-80 Beginners Guide	19

UPGRADES

	16K RAM, 200 nsec, 1 Full Year Guarantee	16
	64K RAM, 150 nsec, 1 Full Year Guarantee	64
26-1122	Model 4 64K RAM Kit with PAL	134
	Same except ours Includes a genuine PAL. Save $$$$	89
26-1123	Model 3 to Model 4 Upgrade Kit	719
26-1125	Model 3 Hi-Resolution Board	314
26-1126	Model 4 Hi-Resolution Board	214
26-1127	Model 4 Drive 0 Kit	551
	Same as above except our brand Save $$$$	369
26-1143	Model 1 Double Density Controller	127
	Famous AEROCOMP "DDC" Double Density Controller	99
26-1145	Model 1 RS232 Kit	85
26-1148	Model 3 RS232 Kit	95
	Same as above except our brand Save $$$$	69
26-1162	Model 3 Drive 0 Kit	550
	Same as above except our brand Save $$$$	369
26-1163	Radio Shack 40 Track Disk Drive	203
	Same as above except 40 track Tandon Save $$$	169
26-4104	Model 2 Hi-Resolution Board	424
26-4105	l64K RAM Board	339
26-4163	8" Disk Drive	510
26-4167	8" Slim Drive	359
	Holmes Mod 1, Mod 3 4Mhz Speedup Mod	89
26-3013	16K RAM For MC-10 Color Computer	42
26-3016	Color Computer Low Profile Keyboard	34
26-3018	Color Computer Extended BASIC ROM	85
26-6010	Model 2/12 to Model 16 68000 Kit	1299
26-6017	Model 12 Card Cage	169

We maintain an extensively equipped service facility with a large stock of parts. Give us a call for that hard-to find item. We'll try and provide what you need

CABLES

We manufacture a great number of different cable assemblies to connect most anything to anything else. Call us with your specific needs. We might have it on the shelf. All at discount prices, of course

RIBBONS

We have more ribbons than you do, so please call us when you need some. Way too many types to list here. They're cheeeep too

SUPPLIES

5.25" SSDD Diskettes, Pack of 10	19
5.25" DSDD Diskettes, Pack of 10	25
8" SSDD Diskettes, Pack of 10	32
8" DSDD Diskettes, Pack of 10	49
5.25" Flipsort, Holds 75 Disks	19
8" Flipsort, Holds 50 Disks	24
8.5" x 11" Tractor Paper Good 20 lb Stock, 2900 Sheets	22
14" x 11" Tractor Paper Good 20 lb Stock, 2900 Sheets	33
5.25" Head Cleaning Kit	9
8" Head Cleaning Kit	9

BUY FROM US RIGHT NOW!

Our large inventory can not be listed completely. Please call if you do not see what you want. Chances are we have it. Because of the time lag in magazine advertising, prices are subject to change without notice. Price reductions will be passed along automatically. Your company and/or personal checks are welcome and will not delay your order. We use TeleCheck. Please observe these requirements and your order will be shipped without delay; The check must be drawn on a USA or Canadian bank and payable in US Dollars. It must be a bank printed check and contain your street address and telephone number. The signature must EXACTLY match the name printed on the check. Thats all there is to it. In addition, we accept American Express, Mastercard, Visa, Cashiers Checks, Wire Transfer, and we also ship COD. COD's and motor freight shipments may require a deposit. Some special items require deposits. ALL COD's require cash or cashiers check on delivery. Company and/or personal checks can NOT be accepted in payment of COD shipments. Credit cards are not charged until we ship your order. Shipping charges are not included in our prices unless otherwise indicated. No State Sales Tax collected on out-of-state shipments or software. If you order by 5PM your order will be shipped the NEXT DAY if stock is on hand. Due to the nature of this business there are NO REFUNDS ON SOFTWARE, REPLACEMENTS WILL BE PROVIDED FOR DEFECTIVE SOFTWARE ITEMS, provided we are notified within ten days of delivery of the merchandise. If you have a particular problem please call us for help and instructions

WARRANTY

All items carry the original manufacturers warranty. We will be happy to mail you a copy upon request

Ad 1083

TeleCheck **Makes checks as welcome as cash.**
COMPANY AND PERSONAL CHECKS WELCOME.
Your order will be shipped immediately upon receipt of your check.

CALL TOLL FREE FROM ANYWHERE IN THE USA

800-527-0347

800-442-1310

VISA MasterCard AMERICAN EXPRESS

Easing Hands-On Computer Time for Teachers

The Teacher Authoring System allows teachers to make use of their lesson preparation skills while not burdening them with computer-related chores.

Reviewed by Mark E. Renne

Many teachers are excited about adding computers to their classrooms, but at the same time, they're worried about the complex problem of programming them for their students. What can be done to solve this paradox?

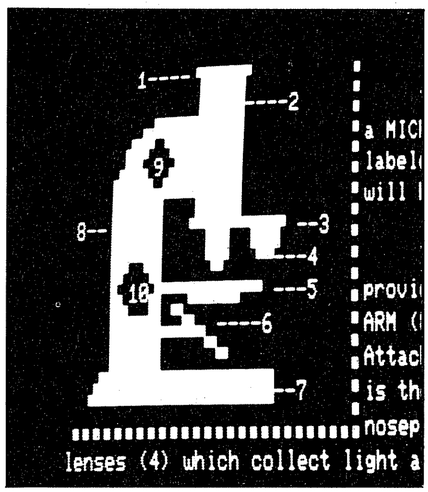
Enter the Teacher Authoring System. This program allows any teacher, even those with no programming experience, to prepare computer-aided lessons for their students to execute. A teacher can now use his or her skills to prepare the content of the lesson and leave the programming to TAS.

Is an authoring system the solution to all your problems? Is TAS the authoring system for you? I'll try to answer those questions as we take a closer look at this new offering from Teach Yourself by Computer Software (TYCS).

First, what is an authoring system? An authoring system allows an instructor to prepare a lesson, including tutorial and questions, for a student to use on a computer. A good authoring system should make use of a teacher's skill in lesson preparation and not bother them with computer-related chores. In other words, the authoring system should serve as a tool for the teacher—an interface between the teacher and the computer. An authoring system will make adding a computer to the classroom easier by allowing the instructor to use the computer without having to become a programmer.

Of course, an authoring system will not solve all the problems of the classroom. In fact, an authoring system, or computers in general, will create a few problems for a teacher. The challenge is to use the advantage of a Computer Assisted Instruction (CAI) system to outweigh the problems associated with putting a computer in the classroom.

The advantage of a CAI program is that it allows students to progress at their



own speed, and it provides the teacher with a complete report of all student activities. For students, the computer offers an opportunity to explore at a higher speed or receive extra help with more difficult subjects.

The disadvantage to a CAI program is that the teacher must be organized and constantly provide students with new, interesting lessons. This may not seem like a disadvantage, but a computer will require even more time from an already harried professional.

Of course, teachers are not the only people who can benefit from a product like the Teacher Authoring System. Retail managers could use it to instruct employees about company policies or new products. Doctors can use an authoring system to provide health information to their patients or their assistants. Chances are that in the future, much of our instruction will come from this type of system. The applications are endless and today's teachers are on the edge of this communication frontier.

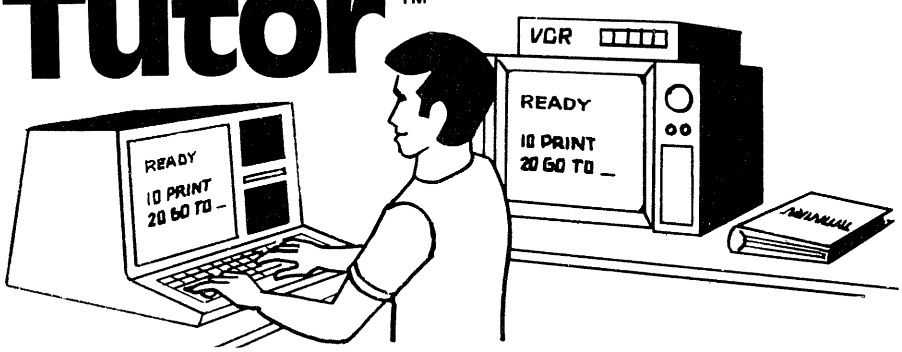
The Teacher Authoring System (TAS) comes with three disks and a manual. There are two "teacher" disks and one "student" disk. The "teacher" or authoring disks come with TRSDOS and are protected from copying. The company has provided two disks, one as backup in case of any media problems. The "student" or presentation disk comes with a sample lesson and is unprotected, allowing you to make as many copies as you have students.

The TAS manual comes in two parts: a tutorial section and a reference section. The tutorial section includes a general discussion of the program procedures and a sample lesson. There are several pages that take you step-by-step through the development of a lesson. This section will probably be most used as you first become acquainted with TAS. The sample lesson, all about using a microscope, is also completely documented with easy-to-understand instructions.

The reference section of the TAS manual is for the more experienced authoring system user. It is arranged in order of the menus that are used throughout the program. Finding information on a particular step is very easy using this program-oriented style. The four sections found in the reference part of the user's manual are authoring, maintenance, presentation, and student reports. These are exactly the steps you would follow to develop your computer lesson. I found the organization of the manual quite pleasing and easy to use. There is a table of contents for each manual, but no index is provided.

Let's take a look at how lessons are prepared using this system. First, you develop the tutorial section of the lesson. This is the written "learning" part of the lesson, which would include a few video pages of text and graphics, if you wish. This part of the lesson is prepared using the text and graphics portions of TAS.

Computer Tutor™



VIDEO INSTRUCTION TAPES

STEP BY STEP INSTRUCTIONS

VHS or BETA FORMAT

PICTURES ARE WORTH THOUSANDS OF WORDS AND SAVE HOURS OF FRUSTRATION

Jse your VCR side by side with your computer o learn disk operating systems, how to program, nd how to use programs. Your VCR along with your omputer serve as your personal tutor. Pause your CR to review and learn at your own pace.

SPECIAL OFFER

azy Writer word processor and video instruction ape complete with documentation and registration ard for model 1, 3 or the model 4, (the model 4 has 0 character display).

ONLY $119.00

VERBATIM DATALIFE

ngle side double density 10 in a hard plastic storage box.

ONLY $25.00

C.O.D. Orders Add $3.00
$3.00 Shipping Per Order
Illinois Residents Add 6% Sales Tax
Specify either VHS or BETA Tape Format

TAPES FOR THE TRS 80

CAT #	TOPIC	APPROX.	PRICE
DOS-1	TRSDOS 1.3 MOD 3	1 HR 30 MIN	$39.95
DOS-6	TRSDOS 6.0 MOD 4	1 HR 45 MIN	$39.95
DOS-3	NEWDOS 80 2.0	1 HR 45 MIN	$39.95
DOS-2	DOSPLUS 3.5	1 HR 45 MIN	$39.95
DOS-5	LDOS	1 HR 45 MIN	$39.95
DOS-7	CPM 2.2 MOD 4 MONTEZUMA	1 HR 45 MIN	$39.95
DOS-4	MULTIDOS	1 HR 45 MIN	$39.95
EW-1	VISICALC (TRS80)	1 HR 30 MIN	$39.95
WP-1	LAZY WRITER	1 HR 45 MIN	$39.95
WP-2	SUPERSCRIPSIT	1 HR 45 MIN	$39.95
WP-3	SCRIPSIT	1 HR 30 MIN	$39.95
WP-4	NEWSCRIPT	1 HR 45 MIN	$39.95
DB-1	PROFILE III PLUS	1 HR 45 MIN	$39.95
DB-2	PFS:FILE	1 HR 30 MIN	$39.95
DIO-3	TRSDOS 1.3 DISK I/O	1 HR 45 MIN	$39.95
DIO-4	TRSDOS 6.0 DISK I/O	1 HR 45 MIN	$39.95
BP-1	LEARNING MOD 3, 4 BASIC	3 HR	$49.95
BP-2	LEARNING MOD 1 BASIC	3 HR	$49.95
TC-1	MTERM	45 MIN	$29.95
UT-1	DOTWRITER	1 HR 15 MIN	$29.95

TAPE FOR THE TRS80 COLOR COMPUTER

BP-5	EXTENDED COLOR BASIC	3 HR	$49.95

TAPES FOR THE COMMODORE 64 AND VIC 20

BP-3	LEARNING C-64 BASIC	2 HR	$49.95
BP-4	LEARNING VIC-20 BASIC	2 HR	$49.95
DIO-1	COMMODORE 64 DISK I/O	1 HR 45 MIN	$49.95
DIO-2	VIC 20 DISK I/O	1 HR 45 MIN	$49.95

TAPES FOR THE IBM PC

EW-2	VISICALC (IBM PC)	1 HR 30 MIN	$39.95
TC-2	MTERM (IBM PC)	45 MIN	$29.95

CIRCLE NO. 28 ON INQUIRY CARD

LYNN COMPUTER SERVICE

6831 West 157th Street Tinley Park, Illinois 60477 (312) 429-1915

master charge THE INTERBANK CARD
BANKAMERICARD VISA

VISICALC IS A TRADEMARK OF VISICORP INC. LAZY WRITER IS A TRADEMARK OF ALPHA BIT COMMUNICATION SCRIPSIT, SUPERSCRIPSIT, PROFILE III PLUS AND TRSDOS ARE TRADEMARKS OF TANDY CORP. LDOS IS A TRADEMARK OF LOGICAL SYSTEMS INC. NEWDOS80 IS A TRADEMARK OF APPARAT INC. DOS PLUS AND MTERM ARE TRADEMARKS OF MICRO-SYSTEMS SOFTWARE INC. MULTIDOS IS A TRADEMARK OF COSMOPOLITIAN INC. VIC 20 AND COMMODORE 64 ARE TRADEMARKS OF COMMODORE BUSINESS MACHINES, INC. IBM IS A TRADEMARK OF INTERNATIONAL BUSINESS INC. CP/M IS A TRADEMARK OF DIGITAL EQUIPMENT CORP. NEWSCRIPT IS A TRADEMARK OF PROSOFT. DOTWRITER IS A TRADEMARK OF RCM COMPUTERS. TM INDICATES A TRADEMARK OF LYNN COMPUTER SERVICE.

The text section is much like a simple word processor; you can move the cursor to various places around the screen and insert text. Any of the Model III characters, including the special ones like spaceships, can be placed on the screen. The graphics portion of the program allows you to include illustrations with your lessons. There are several commands to make graphics creation easy, and you can place graphics anywhere, even within text areas. Both of these functions, text and graphics, have a built-in HELP feature that can be called at any time. After you have finished creating your "video" pages, you save them to a disk file; up to 10 pages may be saved for each lesson.

Now that we've created the lesson, we need some questions to go with it. This brings us to the question development section of the authoring system. Each question contains five lines. The first two lines are for the question itself, and can hold up to 60 characters per line. The third line is for the answer, which can only be typed in uppercase. When the student replies, he or she, too, can only answer in uppercase. This might cause some problems for certain applications, i.e., spelling tutorials.

The fourth line indicates the review session to be referred to if the student enters the wrong answer. This would enable the student to review the tutorial section of the lesson before answering the question again. For most questions, you would have the student review only those sections of the lesson dealing with the question and not the entire lesson. This sequence may contain up to five pages.

The final line of the question section indicates which page, if any, should be displayed with the question. This would be used to ask questions about a particular illustration. You can either create a new text page or use an existing one from the lesson.

Now that we have the text and the questions for the lesson, we need to add the lesson to the lesson directory. The lesson is titled and added to the directory using the directory menu. Finally, a file utilities menu allows you to transfer the files to the student's disk. You have now completed a lesson for students to use. I should note that all the preparatory steps are made using a separate disk from the student disk. The only files on the student disk are the presentation files and the lesson files.

Before presenting a lesson to students, you have several options available to determine how the lesson will work and the type of reports to be generated. You may either allow students to review the material when they miss a question, or present it in a strict "test" format. Questions may be presented in random order, or just as you've written them. You decide how many attempts the student will have at each question and if you'll give the correct answer at the end. Instead of presenting the material as a test, you can also have the material presented in worksheet form.

A report is created each time a student uses the system. This report may be either lengthy or short, depending on your choice. If you wish, a short report will simply save the student(s) name(s), the date, amount of time spent, and which lesson was used for each access of the program. You can also create a report that contains response time and answers given for each question, or one that contains information about total questions completed, number of correct answers, and percent score. Remember, it's up to you how much information to save.

As an authoring system, TAS ranks with any on the market. It is one more example of excellent, affordable software for the TRS-80 market. The documentation is as good as the program. It's actually two sets of documentation: one for the beginner and one for reference. It's easy to use, even for those with little computer experience, and yet it offers all the features of most authoring systems. If you've made the decision to add a computer to the classroom, add the Teacher Authoring System, too, to help you get maximum use from your computer.

Mark Renne has written more than 50 articles and reviews for various computer magazines. He has been involved in the computer industry six years and teaches a class in computer literacy.

Product:
Teacher Authoring System

Manufacturer:
Teach Yourself by Computer Software
2128 W. Jefferson Road
Pittsford, NY 14534

System Requirements:
TRS-80 Model III or 4
minimum one disk drive
minimum 48 KB RAM

Special Requirements: None

Price: $149.95

"THE RESULTS ARE IMPRESSIVE..."
—Dennis Kitsz, 80 Microcomputing; 12/82

Langley-St. Clair's* *Soft·View*™ Replacement CRT's eliminates the strobe, flicker and fatigue from TRS-80's.™

Now you can upgrade your monitor with the new medium persistence green or amber phosphor tube.

State-of-the-art systems such as IBM™ and Apple III™ do not use the less costly "P4" B&W display tube because it is actually intended for TV viewing and its rapid strobes (60 times per second) cause irritating eye fatigue.

No amount of "green plastic" will solve this problem. But the new ***Soft·View*** CRT display tube from Langley-St. Clair will.

- Available in slow decay Green or medium decay "European Amber" (the standard in Europe)
- Made with Lead/Strontium impregnated glass that stops X-ray emission.
- Of high-contrast face glass that also stops most U.V. radiation.
- Available in frosted glass with extra Anti-Glare benefits.
- Easily installed...comes with pre-mounted hardware.
- Warranted for one full year against manufacturing defects or tube failure.
- The finest quality double-dark glass phosphor fields to produce dramatic contrast.
- Ideal for Word Processing and Programming, yet fast enough for Games and Graphics.

LSIS *Soft·View*™ CRT'S

☐ #GN42	Green Phosphor	$79.95
☐ #GN42G	Green Phosphor w/Anti-Glare	$89.95
☐ #OR34	Amber Phosphor	$89.95
☐ #OR34G	Amber Phosphor w/Anti-Glare	$99.95
also available:		
☐ #R22G	Red Phosphor w/Anti-Glare	$139.95
☐ #B22G	Blue Phosphor w/Anti-Glare	$139.95

Plus: $7.00 for packing and UPS Shipping
$17.00 for Overseas, Parcel Post or UPS Blue Label
Add Sales Tax where applicable.
(Inquire about the CRT's we have available for many other computer models)

For MasterCard and Visa Orders only, call
800/221-7070 **(in N.Y. call 212/989-6876)**

Langley-St. Clair Instrumentation Systems, Inc.
132 West 24th St., New York, N.Y. 10011

Actual unretouched photo.
* World's largest supplier of upgraded replacement CRT's.
Soft-View, IBM, Apple and TRS-80 and trademarks of LSIS, IBM, Apple Computer and Tandy Corp.

CIRCLE NO. 25 ON INQUIRY CARD

Too much of a good thing.

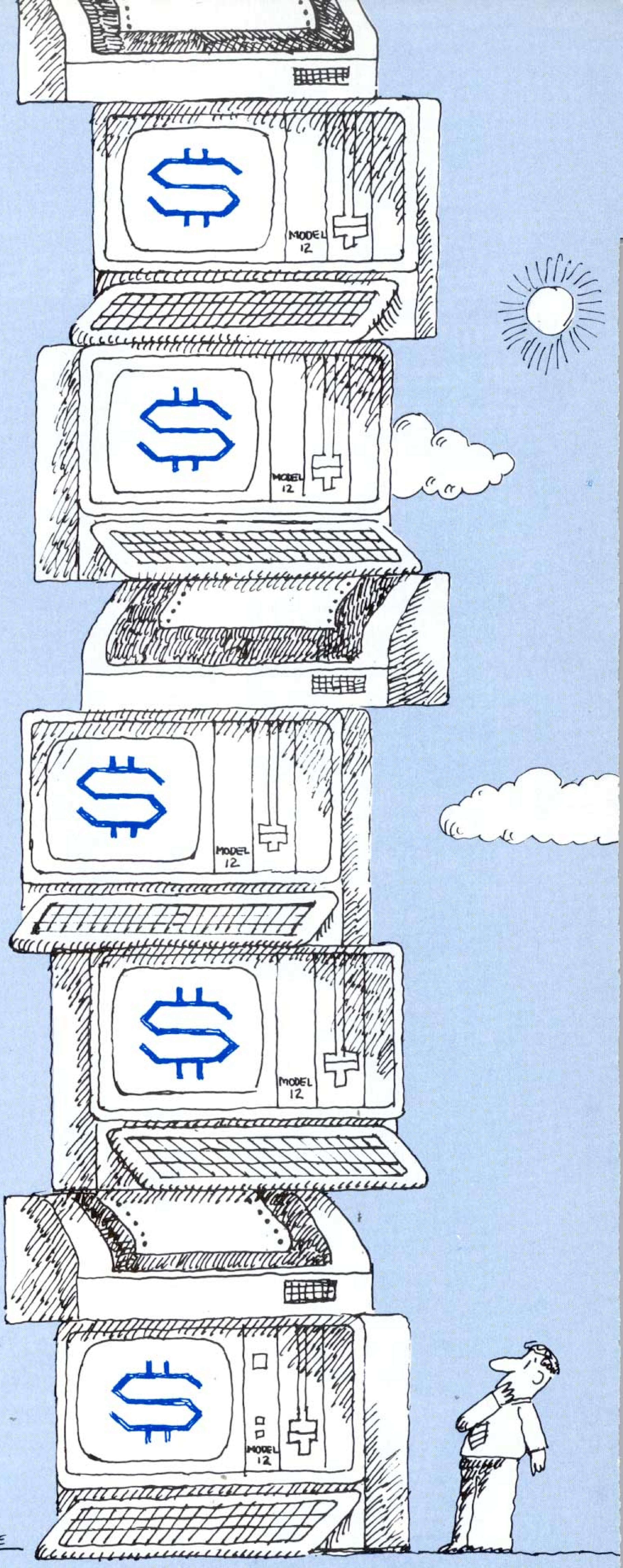
Your TRS-80® is a good thing. It saves you time and money, and it helps your business grow.

But with growth, your data and word processing needs expand. You can buy more computers and spend a lot of money. Or, if you own a Model II, 12, or 16, you can make a good thing better...

OASIS,™ the multiuser operating system from Phase One Systems, can transform your computer into a sophisticated timesharing system. With OASIS, standard terminals, and additional random access memory, three people can use a single TRS-80 at the same time.* Word processing, spreadsheet, and other programs can all run without running into each other. And since OASIS users can share the same data base in a secure environment, up-to-date information is available to everyone. It's an efficient, economical way to grow.

Most importantly, OASIS opens a doorway to hundreds of high performance, multiuser business application packages. Developed, tested, and proven in the business world for more than five years, OASIS applications let you grow and keep you productive.

You can build your OASIS system with components from your Radio Shack® Computer Center. And, if you're just getting started in business computing, we offer a fully compatible single-user version of OASIS. To find out more, call or write us at Phase One Systems, Inc.



PHASE ONE

Phase One Systems, Inc.
7700 Edgewater Drive, Suite 830
Oakland, California 94621
Telephone (415) 562-8085
TWX 910-366-7139

CIRCLE NO. 38 ON INQUIRY CARD

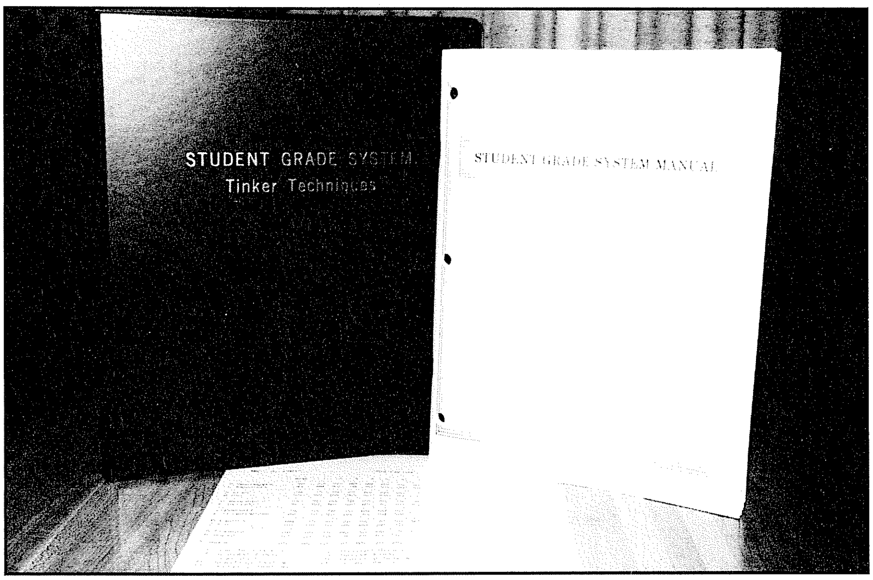
OASIS. It makes a good thing better.

*For optimum system speed, use a hard disk drive. OASIS is a trademark of Phase One Systems, Inc. TRS-80 and Radio Shack are registered trademarks of Tandy Corp.

The Ultimate Teacher's Assistant

The Student Grade System is an easy-to-learn, well-written program that can do most of the record-keeping chores that teachers face.

Reviewed by Hardin Brothers



If I had 40 hours in every day, I probably still wouldn't have time enough for all the things I would like to do with my computer. One project that I've been meaning to do for a long time is to write a grade book program to store and average my students' grades.

Tinker Techniques' Student Grade System has done the job for me with a professional, easy-to-use program that does most of the record keeping I want. I'd like a few more features than this package offers, but the presentation, operation, and thought that have gone into this package are truly impressive.

The program comes in a handsome three-ring notebook with a 64-page manual, including a complete index and a plastic disk holder. The policy statement on the first page of the manual is one of the nicest paragraphs I have read in any software documentation:

"I cannot guarantee that you will like the program, nor can I guarantee that it does exactly what you want it to do... If you are not satisfied, return all materials in their original condition within two weeks, and I will refund your purchase price minus $10.00 for handling."

I wish more software writers would share this confident, fair attitude.

The Student Grade System program, which will run under TRSDOS 1.3 on either a Model III or Model 4, is distributed on a data diskette. Though you will only need one drive to run it, you will need two drives (or a copy utility) to backup the original diskette or to move the program onto a system diskette. The program is written entirely in machine language for speed and security, and because it includes several disk access routines, it will run properly only under TRSDOS 1.3.

The manual is usually thorough, and leads you easily through the process from setting up an electronic grade book to printing out the final results. Once you get going, you'll find that the process of adding new assignments and grades to the system has been carefully planned to require as few keystrokes as possible and to prevent you from making

any obvious or catastrophic errors. You'll be able to enter the day's grades for any class in only a minute or two, and change any previous grades you wish.

The first time you run the Student Grade System (SGS), the program prints the date, a copyright notice, and the message "Hello. Who are you?" You answer with a three-letter (or less) set of initials, which will be used by the system as an extension on your grade book filename. For example, if you enter HIB, the computer will store your grades in the file GRADES/HIB. If you enter more than three characters, only the first three are used. If the program can't find your grades file on the disk, it then asks you for your full name, which will show up on your printed reports and is kept in your file on the disk.

Next, you are prompted for an access password. You define the password when you first use a file. Every other time you attempt to use that file, the program will keep you out unless you know the password. As you type the password in, nothing appears on the screen (someone looking over your shoulder will not be able to learn your password). You have three tries to type in the password before SGS assumes you don't have the right to be using the file and returns to DOS READY. If you are using the program at home and are not concerned about file security, you may just press ENTER for your password without typing anything else.

Next you are shown the program's main menu, and allowed one of six choices:

1. Update a class list
2. Print a class list
3. Update the grade book
4. Print the grade book
5. Set program codes
6. Exit the program (and update disk files)

Oct. 3, 1983 — 1st Period — Teacher's Name

	AVG	1	2	3	4	5	6			7	8	9	10	11
		12	13	14	15	16	17	18	19	20			21	22
		23	24	25	26									
A BERNHARD	83	-	95	85	95	95	-			0	100	100	100	90
		100	70	100	100	73	100	86	100	96			73	67
		92	60	60	78									
M BUCKLEY	67	90	80	78	80	80	68			95	90	90	78	81
		91	95	77	63	20	86	50	0	0			70	56
		49	64	28	76									
S CROSLAND	72	70	85	68	0	0	60			95	100	100	95	76
		91	100	100	81	80	93	0	89	72			97	67
		92	67	88	78									
L DANIELS	81	100	58	85	-	0	50			100	70	100	88	95
		-	90	100	94	47	93	93	86	100			93	70
		97	82	95	97									
B EPSTEIN	72	40	0	0	95	80	0			95	100	60	78	100
		100	90	91	100	100	97	79	75	100			97	93
		70	87	78	86									
A HOCHREITER	78	90	85	75	95	85	75			100	80	90	80	81
		100	95	100	65	40	95	95	86	88			90	89
		0	91	0	97									
J HUTCHINSON	81	90	70	80	80	80	58			100	100	40	90	90
		91	100	100	94	80	97	79	75	84			100	74
		70	76	50	89									
D MARLINI	85	100	88	85	78	70	78			100	80	80	70	90
		82	95	100	88	87	90	86	93	100			83	70
		84	89	78	95									
C MONDAY	94	100	88	88	95	90	90			100	100	90	98	100
		91	100	95	88	87	97	100	96	92			97	81
		97	95	95	97									
C NEELD	50	30	58	78	0	0	0			0	100	90	93	81
		50	95	74	72	27	100	0	0	0			93	63
		68	75	40	46									
M OLANDER	83	100	58	78	58	85	0			100	100	90	98	100
		100	85	100	100	67	86	86	89	92			90	96
		78	89	93	89									
L SIMMS	65	60	58	88	-	-	-			100	90	100	90	90
		72	90	91	69	0	79	64	86	92			0	78
		59	80	0	78									
P SPINOLA	54	40	0	0	0	0	0			0	90	90	90	81
		91	95	100	81	53	97	79	0	92			93	78
		73	64	50	95									
J SYMONS	84	90	75	75	95	85	70			100	70	100	98	71
		100	85	91	69	33	97	79	82	100			100	96
		95	58	83	92									
S WALTON	95	100	95	90	95	90	90			100	80	90	95	100
		91	100	100	88	87	100	100	96	100			100	96
		100	87	100	97									
T WILLS	82	60	88	80	70	80	58			100	100	80	95	86
		100	100	100	88	87	90	79	59	92			83	74
		92	76	68	84									

1. HOUSE OF USHER
2. NOVEL SUM.-CONTENT
3. NOVEL SUM.-GRAMMAR
4. QUOTES-CONTENT
5. QUOTES-GRAMMAR
6. NOTE CARDS
7. QUIZ - NOUNS
8. QUIZ - VERBS
9. QUIZ - ADVERBS
10. QUIZ - ADJECTIVES
11. QUIZ - PRONOUNS
12. VOCAB. QUIZ #1
13. VOCAB. QUIZ #2
14. VOCAB. QUIZ #3
15. VOCAB. QUIZ #4
16. QUIZ-DESCENT...
17. VOCAB. QUIZ #5
18. VOCAB WKSHT 12-13
19. VOCAB WKSHT 14-16
20. VOCAB WKSHT 17-18
21. SPELL TEST 11,I
22. SPELL TEST 11,II
23. SPELL TEST 12-13
24. ROMANTICISM TEST
25. SPELL TEST 14-16
26. SPELL TEST 17-18

SGS numeric class grades

Assuming that you are using the program for the first time, you would select option 1 to enter the names of students in your class.

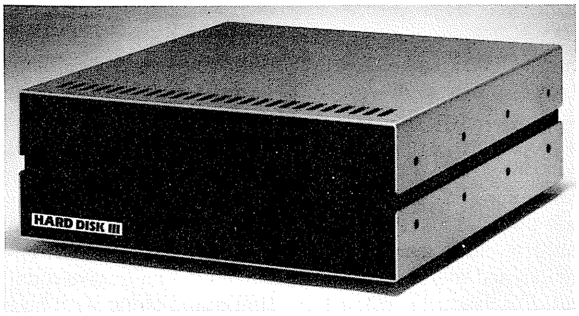
You are shown a new menu and asked to enter a class code (a single letter or number to identify the class, e.g. a period number). You are then asked to type in the name of the class and the grading method you will use. Once you select the grading method, you won't be able to change it, because all the future programming parameters depend on your selection. You may chose among numeric grades, A - D & F, A - F (including E), A – D & F with +'s and –'s, or A – F with +'s and –'s. You may choose different grading systems for different classes if you wish.

Next, you are asked to enter students' names. Though you may enter as many names as you wish, the screen will only show 28 at a time. While you are entering the names, you may become aware of one of SGS's limitations: first names can only be six letters long and last names can only be 12 letters long. If you exceed these limits, the extra characters will be dropped by the system.

Once you've entered the names, the system asks you if everything is okay. If you answer yes, you are then asked if you wish to alphabetize the class list. If your regular class lists are in alphabetical order, you will probably answer yes. If you have extra students added at the

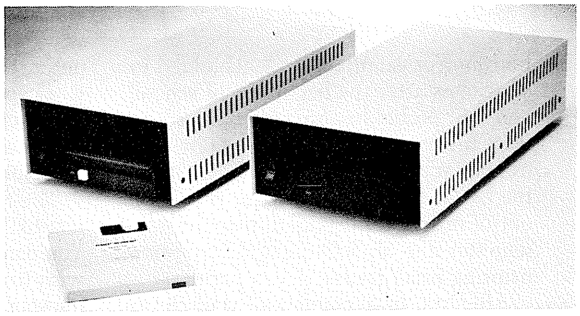
EXTRAORDINARY VALUES

PANTHER



5 MEG	$1995.00
5 MEG + 5 MEG REMOVABLE	2895.00
10 MEG	2295.00
10 MEG + 10 MEG	3495.00
10 MEG + 5 MEG REMOVABLE	3395.00
15 MEG	2395.00
15 MEG + 15 MEG	3795.00
15 MEG + 5 MEG REMOVABLE	3495.00
Underdesk mounting bracket	34.95
Boot Rom (Model III & 4)	34.95

PANTHER CUB



5 MEG FIXED *MASTER	$1795.00
5 MEG FIXED +SLAVE	1349.00
5 MEG REMOVABLE *MASTER	1995.00
5 MEG REMOVABLE +SLAVE	1495.00
Cartridge	99.95

* Master Drive includes power supply, disk controller, I/O adapter, enclosure and applicable cabling.
+ Slave Drive includes power supply, enclosure, connector cabling.

ADAPTER MODULES

IBM PC/XT • Apple • ComPAQ • LNW-80, II • Franklin • TRS-80 Model I, III, 4

Coming soon Xerox 820 — II adapter. • 180 day warranty • Information available upon request.

CALL FOR YOUR LOCAL PANTHER/PANTHER CUB DEALER.

Model 4 $1699.

128K 2 drive, VR-RS232 Sound generator, TRS-DOS & manual

Disk III kit

Internal floppy disk subsystem for Model III & 4 (includes controller, power supply, cabling, bracket and all associated mounting hardware).

DISK III KIT — Without Drive	$299.95
DISK III KIT — With 1 drive (ss,dd)	449.00
DISK III KIT — With 2 drives (ss,dd)	699.00

External Drives includes power supply, enclosure and extender cable	$99.00
2 drive cable	25.00
VR-RS232	80.00

SPECIAL BUYS ON COD & PREPAID ORDERS ONLY:

Epson RX-80	$399.00
Epson RX-80FT	425.00
Epson FX-80	650.00
Epson FX-100	850.00
Epson MX-100	750.00
Epson Ribbons	Call
Epson Graftrax	75.00
Amber Screens	99.00
Smith-Corona Printwheels	$4.98
Smith-Corona Ribbons	4.25
Spinwriter Multistrike Ribbons	4.95
C. Itoh & Diablo Ribbons	7.50
Verbatim Diskettes 525	26.90
Nashua 8″ Diskettes	42.95
8″ Library Cases	4.00
Generic Diskettes	$1.75
Green Screens	99.00
Percom Doubler	169.00
DC Hayes Smartmodem 300 baud	229.00
DC Hays Smartmodem 1200 baud	569.00
Paper 9½″ x 11″ fanfold	29.95

Call Toll Free • 800-345-8102

Published prices reflect cash discount. All prices are subject to change without notice. TRS-80 and TRSDOS are trademarks of Tandy Corp.

Telephone Hours: 8:30am—7pm MON-FRI., SAT 10-3 EST
CABLE "VRDATA" TELEX 845-124

VR data

(215) 461-5300
777 Henderson Blvd.,
Folcroft, PA 19032

VR DATA — WEST
WATONGA, OK
1-405-623-8664

bottom (for example, students added to your class after the beginning of school), you will probably answer no, in order to keep the computer file in the same order as your roll book. You can alphabetize the list at any time you wish, keeping it out of order on purpose during the semester and alphabetizing it just before printing final grades, if you prefer.

After you have entered all your classes, you may return to the main menu. You may wish to print out your class lists (option 2) next. Such lists are handy for field trips and special assignments, or any other special uses you may have in mind.

The third option on the main menu, Update grade book, takes you to the heart of SGS. You will first want to enter information about your assignments. Each assignment can be described with a maximum of 20 characters, and put into one of eight categories: Classwork, Exam, Homework, Paper, Quiz, Review, Test, or Other. Each assignment can also be given a relative weight between 1 and 5. If you are using a numeric scoring system, you will be asked to enter the maximum score of each assignment.

Once you have defined the assignments, you return to the class list and enter grades for each student. You never have to press the ENTER key while recording scores; you are automatically moved to the next blank after you finish filling up the grade field. For example, if you are using numeric scores and an assignment has a maximum of 50 points, as soon as you enter two digits, SGS will move the cursor to the next grade field.

SGS is very flexible. At any time, you may change a student's score on any assignment, change the relative weight or category of any assignment, delete an assignment, or add more assignments to the file. The only limitations are memory size and an SGS requirement that no more than 45 assignments be recorded for any class (if you need more, you may handle the class as two separate classes and average the final grades for each section by hand).

If you return again to the main menu, you can select option 4 to send your grade book, complete with current student averages, to your printer. Before printing the grades, SGS asks you to give relative weights to the eight possible categories of assignments. You can link categories together if you wish, ignore certain categories, or weight the categories in any manner as long as the total weights add up to 100 percent. If you have a printer attached to your computer, SGS then sends the grade book to it. If you don't, it displays every student's current average on the screen. You may either print class averages or a complete record of each class.

The fifth main menu option allows you to set certain program parameters. You can set whatever characters you wish to indicate excused scores and unexcused scores. For example, when I use letter grades for a class, I select a dash (—) to indicate an excused score

The SGS system is quite security-conscious. If you don't know the correct password, you are kept from viewing or changing files. All information is coded before it is sent to disk to prevent easy tampering.

and zero (0) to indicate an unexcused, undone assignment. You can also select whether you wish the keyboard to be in all uppercase mode or mixed-case, typewriter style.

The fifth menu option also allows you to set printer parameters. You can select the maximum number of characters per line, the length of each page, and printer control codes. If your printer is capable, you may decide to print your grades in emphasized mode, or with condensed characters, or even set left and right margins. You will have to read your printer manual to find out what codes to set in SGS.

The final main menu option allows you to exit SGS gracefully and save all the changes you have made on disk. SGS saves all files twice, allowing you to easily backup all your class data. If you wish, you may also leave SGS without saving any changes to disk.

The SGS system is quite security-conscious. If you don't know the correct password, you are kept from viewing or changing files. All information is coded before it is sent to disk to prevent easy tampering. Finally, when you exit SGS, it erases all of the memory it has used, so no one can read your grade information with a utility like Debug.

Along with all its positive features, SGS also has some limitations. First, even if you have 48 KB RAM in your Model III or 4, SGS will not use the top 16 KB. Since it saves and loads your complete grade book for all your classes at once (unless you define each class with a separate three-letter code), this can restrict you unnecessarily.

Second, regardless of available memory, SGS will not allow more than 45 grades per class. In some of my classes, I like to give a daily participation grade along with homework and classwork grades. SGS will not handle the necessary number of grades for this kind of system.

Third, SGS will not print full-class reports to the screen. If you want to review all the grades for a whole class, you must send them to a printer.

Fourth, once or twice a quarter, I like to give each student a complete record of his or her work to date. I also like to have such reports available for parent or counselor conferences. SGS will not generate such a report, though you can use the TRSDOS screen-dump command, <SHIFT> <down-arrow> <*>, to get around this limitation. However, the individual reports will not have the professional look of SGS's regular printouts.

Finally, SGS performs all grade figuring and rounding without reporting how close students are to the border line between grades. If you use a numeric grade book, it will only report a student's average, not a total score. If you use letter grades, it will decide when to round up or down. The subjective, human factor that many teachers like to use when determining final grades is more difficult with SGS than with other systems.

Despite these limitations, if SGS's capabilities match your grade-keeping methods, I think you'll find it a most worthwhile program. It is easy to learn and use, well-written, and seems to be thoroughly debugged. And, because it is written in machine language, it is also fast; it can alphabetize an entire class in only one or two seconds.

Product:
Student Grade System

Manufacturer:
Tinker Techniques
435 Greenway Avenue
Trenton, NJ 08618

System Requirements:
Model III or 4
minimum one disk drive
(cassette version also available)

Special Requirements:
A printer is recommended for full use of software

Price: $75

AN APPLES AND ORANGES SITUATION?

by Hardin Brothers

The Model 4 and Apple IIe may be essentially identical in base cost, but the similarities end there. The following is a close look at the two systems—their strengths and weaknesses.

I spend at least three hours a day working with my TRS-80 Model 4—writing, programming, and learning. I spend another two hours each day with a roomful of Apple IIe's, teaching high school students to program in Basic. Because of my experience with both computers, I have compiled, mostly for my own use, a list of differences between the two machines.

Most people who own a microcomputer seem to become very attached to their own brand and look down on others. I am no exception, except that now, with my experience with both computers, I have the background to support my feelings. If you are thinking of buying either a TRS-80 or an Apple, read on before taking the plunge.

The first question my non-computing friends always ask about a computer is, "How much does it cost?" At first glance, it appears that the Apple and the TRS-80 are essentially identical in cost. An Apple IIe starter kit lists for $1,995. A Model 4 is $1,999.95. But the base cost doesn't tell the whole story.

The Model 4, for that price, is a 64 KB RAM computer with a monochrome monitor, dual 5-1/4" disk drives, an internal speaker, a cassette port, a parallel printer port, an RS-232C port, and a disk port for attaching two more drives. It also has a bus-expansion port, which allows peripheral equipment to be attached directly to the computer's address and data lines.

The Apple IIe starter kit includes a 64 KB RAM computer, a monochrome monitor, a single disk drive and disk drive card, an 80-column card, a cassette port, an internal speaker, and seven internal expansion card slots. To upgrade the Apple into a machine more comparable to the Model 4, you would need to add another disk drive ($395), and a parallel printer port ($165).

The two machines are still not exactly the same. If you wish to add a third or fourth drive to the Apple, you have to buy another disk drive card, but if you wish to add additional disks to the TRS-80 you need only plug them in. Our Apple, which now costs $2,555, still doesn't have a serial (RS-232) port, but if you plan to use it for connecting a modem, you'll probably find that an internal, complete modem for the Apple costs about the same as an external one for the TRS-80. And, the Apple for that price is not capable of displaying the color graphics for which Apple is acclaimed.

Photos courtesy of Apple Computer and the Tandy Corp.

Of course, $2,555 for the Apple and $1,999.95 for the Model 4 are both "suggested retail." Apples are extensively discounted by independent dealers; discounted TRS-80's are also available, but more difficult to find except by mail order. Both Apple and Radio Shack offer school discounts.

The TRS-80 is built into a single cabinet that holds the keyboard, monitor, and disk drives. When you bring it home and open the box, all you need to do is slide it out and connect its single power cord to a wall outlet. The Apple is modular in design. You take the computer (keyboard and case) out of one box, the video monitor out of another, the 80-column card out of a third, and the disk drive and controller card out of a fourth. Before you can turn it on, you need to open the computer case, thread the disk drive cable through a slot in the back, hook it to the drive-controller card, insert the card into an internal slot, and clamp the wire in place with a special connector. You also have to slide the 80-column card into its special internal slot. I have a fair amount of mechanical aptitude and am no stranger to the insides of computers—it took me about 45 minutes to set up an Apple IIe the first time I tried.

Once the machines are set up, the first thing you might notice is the difference in the keyboards. The Apple keyboard has a number of special symbols which are apparently missing on the Model 4: (,), ^, \ , [,], :, and two keys called Open Apple and Closed Apple. However, all of the special symbols are available on a Model 4 as two-key combinations—if you use an application or language program that needs them, you will soon learn to find them easily.

On the other hand, the Model 4 has three programmable keys (F1, F2, and F3) and a separate numeric keypad; the Apple has neither. If you are going to use the computer for any number-intensive purposes (e.g., using VisiCalc or averaging grades), a keypad will be far more useful than the special symbols. An external numeric keypad is available for the Apple at an additional cost.

As you unpack the Apple, you will notice that each piece has its own manual and installation instructions. The curious thing is that none of the Apple manuals contain any indication of the Basic commands that are available. If you wish to program in Basic, and want a complete instruction set, you must purchase a separate $50 *Apple Programmer's Guide*. You might get around that cost by buying one of

Basic Command	Disk Basic Model III Mode	Disk Basic Model 4 Mode	Applesoft & DOS 3.3
Variable Types			
String ($)	Yes	Yes	Yes
Integer (%)	Yes	Yes	Yes*
Single (!)	Yes	Yes	Yes*
Double (#)	Yes	Yes	No
Algebraic Operators +, −, *, /, ^	Yes	Yes	Yes
Relational Operators =, <, >, <=, >=	Yes	Yes	Yes
Logical Operators AND, OR, NOT	Yes	Yes	Yes
Boolean Operators AND, OR, NOT	Yes	Yes	No
General Statements			
AUTO	Yes	Yes	No
CALL	No	Yes	Yes*
CHAIN	No	Yes	Yes*
CLEAR	Yes	Yes	Yes*
CLS	Yes	Yes	Yes
COMMON	No	Yes	No
CONT	Yes	Yes	Yes
DATA, READ	Yes	Yes	Yes
DEFDBL, DEFSNG DEFINT, DEFSTR	Yes	Yes	No
DEF FN	Yes	Yes	Yes
DEF USR	Yes	Yes	No
DELETE	Yes	Yes	Yes
DIM	Yes	Yes	Yes*
EDIT	Yes	Yes	No*
END	Yes	Yes	Yes
ERASE array	No	Yes	No
ERL	Yes	Yes	No
ERR	Yes	Yes	No
ERROR	Yes	Yes	No
FOR/NEXT/STEP	Yes	Yes	Yes
GOSUB	Yes	Yes	Yes
GOTO	Yes	Yes	Yes
HTAB	No	No	Yes
IF...THEN	Yes	Yes	Yes
IF...THEN...ELSE	Yes	Yes	No
INP	Yes	Yes	No
INPUT	Yes	Yes	Yes
LINE INPUT	Yes	Yes	No
LIST	Yes	Yes	Yes
LLIST	Yes	Yes	No*
LOAD	Yes	Yes	Yes
LPRINT	Yes	Yes	No*
MERGE	Yes	Yes	No*
NEW	Yes	Yes	Yes
ON ERROR GOTO	Yes	Yes	Yes
ON...GOSUB	Yes	Yes	Yes
ON...GOTO	Yes	Yes	Yes
OPTION BASE	No	Yes	No
OUT	Yes	Yes	No
PEEK, POKE	Yes	Yes	Yes
POP	No	No	Yes
PRINT	Yes	Yes	Yes
PRINT USING	Yes	Yes	No
PRINT @	Yes	Yes	No
PRINT TAB	Yes	Yes	Yes

Table 1. *Comparison of Model 4 and Apple Basics (Note: when similar functions have different names, the TRS-80 function names are used.)*

Basic Command	Disk Basic Model III Mode	Disk Basic Model 4 Mode	Applesoft & DOS 3.3
REM	Yes	Yes	Yes
RENUM	Yes	Yes	No*
RESTORE	Yes	Yes	Yes
RESTORE n	No	Yes	No
RESUME	Yes	Yes	Yes*
RETURN	Yes	Yes	Yes
RUN	Yes	Yes	Yes
RUN, R	No	Yes	Yes
SAVE	Yes	Yes	Yes
SAVE, A	Yes	Yes	No
SAVE, P	No	Yes	No
SOUND	No	Yes	No
SPEED	No	No	Yes
STOP	Yes	Yes	Yes
SWAP	No	Yes	No
SYSTEM	Yes	Yes	No
TRON, TROFF	Yes	Yes	Yes
USR	Yes	Yes	Yes*
VTAB	No	No	Yes
WAIT	No	Yes	Yes
WHILE. . .WEND	No	Yes	No
WRITE	No	Yes	No
String Operations			
ASC	Yes	Yes	Yes
CHR$	Yes	Yes	Yes
DATE$	No	Yes	No
ERR$	Yes	Yes	No
HEX$	No	Yes	No
INKEY$	Yes	Yes	No
INPUT$	No	Yes	Yes*
INSTR	Yes	Yes	No
LEFT$	Yes	Yes	Yes
LEN	Yes	Yes	Yes
MID$ =	Yes	Yes	No
= MID$	Yes	Yes	Yes
OCT$	No	Yes	No
RIGHT$	Yes	Yes	Yes
SPACE$	No	Yes	No
SPC	No	Yes	Yes
STR$	Yes	Yes	Yes
STRING$	Yes	Yes	No
TIME$	Yes	Yes	No
VAL	Yes	Yes	Yes
Arithmetic and Other Functions			
ABS	Yes	Yes	Yes
ATN	Yes	Yes	Yes
CDBL, CINT, CSNG	Yes	Yes	No
COS	Yes	Yes	Yes
EXP	Yes	Yes	Yes
FIX	Yes	Yes	No
FRE	Yes	Yes	Yes
INT	Yes	Yes	Yes
LOG	Yes	Yes	Yes
LPOS	No	Yes	No
MEM	Yes	Yes	No
POS	Yes	Yes	Yes
RANDOM	Yes	Yes	Yes*
RND	Yes	Yes	Yes*
ROW	No	Yes	No
SGN	Yes	Yes	Yes
SIN	Yes	Yes	Yes

***Table** (continued)*

the many books available that lead the reader through learning Basic on an Apple, but none that I've seen has a complete description of the commands available in Applesoft Basic.

The Model 4 comes complete with a manual that details both TRSDOS 6.0 and its accompanying Microsoft Basic. This manual is not one of Radio Shack's better instructional manuals; in fact, it is enough to scare off any first-time user. A separate book, which is much clearer and friendlier, is available from Radio Shack—it explains how to use the optional ROM Basic.

Both machines have a version of Basic available in ROM. The Apple has a second, smaller version of Basic, "Integer Basic," available that may be loaded into the computer from the DOS diskette.

The Model 4 can run three different versions of Basic: its resident ROM Basic (which is identical to the ROM Basic that was available on a Model III), Model III Disk Basic under TRSDOS 1.3, and a full-blown Mbasic under TRSDOS 6.0. Figure 1 is a comparison of commands available under both TRS-80 Disk Basics and Apple's Applesoft Basic running with DOS 3.3 installed.

Table 1 was difficult to compile. Often the same commands in different versions of Basic or DOS have dissimilar functions. Also, similar functions are often invoked by different names. I have used TRS-80 nomenclature throughout the list and matched Apple capabilities as much as possible. An asterisk (*) indicates a significant difference in power between at least two of the systems, which is explained in the notes at the end of the list. I believe the list is accurate, but I may have overlooked some subtleties along the way. If so, I apologize to whichever machine I inadvertently slighted.

One type of command is conspicuously absent from figure 1; I have purposely omitted those Basic commands that deal solely with graphics. One of Apple's strong points is supposed to be its graphics capabilities. It can display both high-resolution and low-resolution graphics in several colors. The TRS-80 has an undeserved reputation of having poor graphics. Consider these points:

1. The normal TRS-80 graphics have three-times higher resolution graphics than Apple's low-resolution graphics (six pixels per character space as opposed to two).

2. If you add the high-resolution graphics option ($249.95) to a TRS-80, you have control over 153,600 pixels as opposed to 44,800 available in Apple's high-resolution mode.

3. The TRS-80 high-resolution

graphics option includes 32 KB of extra RAM; the Apple graphics mode uses up to 16 KB of regular memory.

4. When you use the normal TRS-80 graphics, you can mix normal characters and graphics anywhere on the screen. The TRS-80 character set includes 95 special characters as well as Japanese Kana characters (a feature that is not very useful for most of us). When you use Apple graphics, you may only include text (and no graphics) on the bottom four screen display lines. Text and graphics can never be mixed. Also, the Apple does not have access to any special characters, though shapes of any sort may be defined in high-resolution graphics.

5. Apple graphics may be displayed in color on your home TV. If you intend to use the 80-column feature, however, you will have to purchase an expensive color monitor to include both 80-column text and color graphics in the same display screen. 80-column text is completely blurred and illegible on a normal color TV.

6. On the other hand, Apple's high-resolution graphics commands are included in its standard language. Also, a large number of commercial programs have been written to support Apple graphics; almost none have been written for TRS-80 high-resolution graphics.

In the Model 4 running under TRSDOS 6.0, the 80-by-24 screen does not use any of the 64 KB RAM of the computer. Instead, a specially addressed bank of 2 KB RAM handles screen memory. All computer functions work normally with this display and can be shifted easily to 40 characters per column if need be.

The Apple uses some of its RAM for its normal 40-by-24 screen display. When you switch on the 80-column display, the left half of the screen is still stored in memory and the right half is stored on the 80-column card. Because of this split screen memory, several Basic commands—PRINT TAB ( ), HTAB, and comma tabbing in PRINT statements—will no longer work properly. To position the cursor, you have to use either normal PRINT statements or a series of POKES into memory.

Apple's 80-column display card is not completely compatible with either Basic or DOS. The documentation warns that both the Basic RUN command and the DOS CATALOG command "may produce unexpected results" when the screen is in 80-column mode. If the cursor is not on the left half of the screen when you switch from 80 columns to

Basic Command	Disk Basic Model III Mode	Disk Basic Model 4 Mode	Applesoft & DOS 3.3
SQR	Yes	Yes	Yes
TAN	Yes	Yes	Yes
VARPTR	Yes	Yes	No
Disk File Commands			
CLOSE	Yes	Yes	Yes
CVD, CVI, CVS	Yes	Yes	No
EOF	Yes	Yes	No
FIELD	Yes	Yes	No
GET	Yes	Yes	Yes
INPUT#	Yes	Yes	Yes
INPUT$	No	Yes	No
LINE INPUT#	Yes	Yes	No
LOC	Yes	Yes	No
LOF	Yes	Yes	No
LSET	Yes	Yes	No
MKD$, MKI$, MKS$	Yes	Yes	No
OPEN (O,I,E,R)	Yes	Yes	Yes
PRINT#	Yes	Yes	Yes
PUT	Yes	Yes	Yes
RSET	Yes	Yes	No
WRITE#	No	Yes	No
Operating System Command and Utilities			
APPEND	Yes	Yes	Yes*
ATTRIB	Yes	Yes	No
AUTO	Yes	Yes	Yes
BACKUP	Yes	Yes	Yes
BOOT	No	Yes	No
BUILD	Yes	Yes	Yes
CLEAR	Yes	Yes	No
CLICK/FLT	No	Yes	No
CLOCK	Yes	Yes	No
COM/DVR	No	Yes	No
COMM	No	Yes	No
CONV(ERT)	Yes	Yes	Yes
COPY	Yes	Yes	Yes
CREATE	Yes	Yes	No
DATE	Yes	Yes	No
DEBUG	Yes	Yes	Yes*
DEVICE	No	Yes	No
DIR	Yes	Yes	Yes*
DO	Yes	Yes	Yes
DUAL	Yes	Yes	No
DUMP	Yes	Yes	Yes
FILTER	No	Yes	No
FORMAT	Yes	Yes	Yes*
FORMS	Yes	Yes	No
FREE	Yes	Yes	No
HELP	Yes	No	No
JCL	No	Yes	No
JOBLOG	No	Yes	No
KSM/FLT	No	Yes	No
LIB	Yes	Yes	No
LINK	No	Yes	No
LIST	Yes	Yes	No
LOAD	Yes	Yes	Yes
LPC	Yes	Yes	No
MASTER	Yes	Yes	No
MEMDISK	No	Yes	No
MEMORY	Yes	Yes	Yes*
MEMTEST	Yes	No	No
PATCH	Yes	Yes	No

Table (continued)

Basic Command	Disk Basic Model III Mode	Disk Basic Model 4 Mode	Applesoft & DOS 3.3
PAUSE	Yes	Yes	No
PROT	Yes	Yes	No
PURGE	Yes	Yes	No
RELO	Yes	Yes	Yes
REMOVE	Yes	Yes	Yes
RENAME	Yes	Yes	Yes*
REPAIR	No	Yes	No
RESET	No	Yes	No
ROUTE	Yes	Yes	No
RUN	Yes	Yes	Yes
SET	No	Yes	No
SETCOM	Yes	Yes	No
SETKI	No	Yes	No
SPOOL	No	Yes	No
SYSGEN	No	Yes	No
SYSTEM	No	Yes	No
TAPE100	No	Yes	No
TIME	Yes	Yes	No
VERIFY	No	Yes	No*
WP	Yes	Yes	No*

Notes & Comments

■ Applesoft's integer variables, which take the same amount of storage space as floating-point variables, cannot be used as indexes of FOR/NEXT loops, DEF statements, or Boolean operations. They are most useful for increasing access speed to items in an array.

■ Applesoft only recognizes one type of real-number variable; it's between the TRS-80's single and double-precision variables in accuracy.

■ Model 4's CALL allows an unlimited number of arguments to be passed to a machine language subroutine. Applesoft's CALL does not allow any variables to be passed.

■ Model 4's CHAIN allows several options including a MERGE, designation of a line number for beginning of execution, and deletion of selected lines. However, all CHAINed programs must be saved in ASCII format. Applesoft's CHAIN has no options but does not require CHAINed program to be in ASCII format. (Applesoft can only save programs in tokenized form.)

■ All TRS-80 Basics allow the DIM statement to be used with non-subscripted variables as a way of reserving space in the variable tables in memory. Applesoft does not.

■ TRS-80 Basics allow editing any Basic line in memory with a collection of special editing commands. Applesoft allows editing lines only after they have been LISTed to the screen, and requires combinations of the ESC key and arrow keys to position the cursor, change parts of the line, and read other parts of the line into the input buffer. Both systems can be difficult to learn at first.

■ Instead of including special commands to enable printer use (LLIST, LPRINT, etc.), Applesoft includes a single command to turn on or off any input or ouput device and then sends or fetches material from that device until the command is overridden. No special filtering or formatting of output information is possible—your program must do all the calculations for margins, lines per page, top-of-form commands, etc.

■ Applesoft uses a special DOS utility to perform merging. The original program, the RENUMBER/MERGE utility, and the program to be merged must all reside in memory before the MERGE command is given. Therefore, dynamic merging of programs during program execution is almost impossible.

■ Model 4 Basic allows RESUME to any program line number after an error-trap (ON ERROR GOTO). Applesoft only allows a RESUME to the statement where the original error occurred. Many types of error-trapping are therefore impossible in Applesoft.

Table (continued)

40 columns, it can disappear completely into the second half of screen memory.

If there are any real standards for Basic on a microcomputer, they are those of the two most prominent dialects: Mbasic and Cbasic. The TRS-80 Models I and III had a dialect of Basic written by Microsoft (the creators of Mbasic) that was reasonably complete. The Model 4 uses a complete Mbasic.

Applesoft Basic was not written by Microsoft, and, while it is generally a subset of Mbasic, it has some notable differences. You will notice in table 1 and the accompanying notes that some of the advanced features of Mbasic are missing in Applesoft. Many of the missing features can be programmed around, either by writing special subroutines or by using PEEKs and POKEs to specific system values, but the resulting programs then become very machine specific and cannot be transported to other programming environments without extensive rewriting.

I try to encourage my students to write programs with as much structure as possible, though several features of Applesoft make their task more difficult:

1. It is impossible to space commands in Applesoft in any way other than approved by the interpreter. If you try to indent the lines of a FOR/NEXT loop, for example, the interpreter erases all leading spaces before parsing the line. When it is LISTed, the leading spaces are gone.

2. Many programmers regard Basic's GOTO statement as a primary culprit of unstructured programming. However, because Applesoft does not allow an ELSE in an IF. . .THEN statement, and because it does not have a WHILE. . . WEND structure for checking a value at the top of a loop, GOTOs are very difficult to avoid.

3. Applesoft's display and string-handling commands are notably weak. It does not have a PRINT USING command, does not allow MID$ on the left side of an equal sign (=), does not include an INSTR function, and does not allow string-building with STRING$.

Almost all intermediate and advanced Basic programs include file handling of some sort. Applesoft handles sequential files similarly to TRSDOS Basic. The only standard sequential-file command it has omitted is LINE INPUT#. If sequential files will fill your programming needs, Applesoft should provide sufficient power.

Applesoft has notable problems using Random Access files. It does not allow

■ Applesoft Basic, like non-disk TRS-80 Basic, allows only one USR routine to be defined at a time. Model 4 Basic allows both numeric and string variables to be passed with the USR routine. Applesoft only allows numeric variables.

■ Applesoft's GET is similar to the TRS-80 INPUT$, except that GET only waits for a single keystroke while INPUT$ can wait for any number of keystrokes designated by the program. Applesoft has no equivalent to INKEY$.

■ Applesoft constantly reseeds the random number generator, so the RANDOM command is unnecessary. If RND is used with a negative argument (e.g., RND(−3)), a specific repeatable sequence of random numbers will be generated. RND(0) returns the last random number generated and RND(1) returns a random number between 0 and 1. Applesoft does not recognize expressions like RND(10), used on TRS-80s to return a random integer between 0 and 10.

■ The Apple IIe has a built-in monitor program that performs some of the same functions as DEBUG. It will not, however, show an ASCII dump of memory, or read disk tracks like the Model 4 DEBUG.

■ Apple DOS 3.3 has a CATALOG command that is similar to DIR, but it will not tell you creation dates of files, or the amount of space remaining on disk. It does not sort files alphabetically, does not allow partspecs or partial listings of the directory, and does not flag files that need to be backed up.

■ Whenever Apple DOS 3.3 formats a new disk, it must save the entire DOS kernel on the disk as well as a Basic program. Data disks (without DOS) do not exist in the Applesoft world.

■ Apple allows the setting of HIMEM and LOMEM, but does not allow you to clear memory or do any of the other options available from TRSDOS's MEMORY command.

■ When Apple RENAMEs a file on disk, it does not check to see if that file name already exists, so it is quite possible to have more than one file on a disk with the same name. Only one of those files will then be accessible (though you can't be sure which it will be).

■ In Apple DOS, VERIFY means, "Check a disk file to be sure it is readable." Apple DOS does nothing to insure that what you write to a disk is what is actually stored there.

■ Apple DOS is incapable of write-protecting an entire disk, but it can write-protect individual files with the LOCK command.

***Table** (continued)*

record FIELDing, and permits reading only an entire record or a single byte at a time. More importantly, Applesoft does not include any substitute for the LOF and EOF commands, so you must either include an end-of-file marker of your own inside a disk file, or else your program must know exactly how many records to expect. The first option is more efficient (do you really want to keep track of exactly how many names and addresses are in your mailing list program?), but makes APPENDing more information to the end of a file difficult.

As one Apple salesman said to me, "Compared to TRSDOS, Apple's DOS 3.3 is a joke." The first difference between them is the way the two systems handle information on a diskette.

A standard Model 4 includes two 40-track, double-density disk drives. Each can format a disk with 18 sectors on 40 tracks, for a total of 720 sectors. Apple DOS formats disks with 35 tracks of 16 sectors each, for a total of 560 sectors.

TRSDOS can create two types of diskettes: system disks, which hold the DOS and various utilities, and data disks, which contain programs and other files. Apple DOS can only create system disks. Every disk used on an Apple under DOS 3.3 must contain the DOS on tracks 0, 1, and 2, as well as a directory on track 17. Therefore, the user is restricted to 496 sectors of storage space on every disk. A two-drive Apple can access a total of 248 KB of user storage at any one time.

A Model 4 data disk, after allocating the necessary 24 sectors for the boot code and directory, has 696 sectors available for the user. The amount of space available on the system diskette depends on how many system utilities you want available. A standard configuration that allows everything you need to program in Basic and run /CMD programs leaves 348 free sectors. Together, there are 1,044 sectors, or 261 KB available on the two drives.

Because Apple DOS is much smaller than TRSDOS, it is a better choice for Those who are restricted to a single drive. With a dual-drive system, however, TRSDOS allows more information to be held on-line at any given time. TRSDOS also allows for easy expansion.

One handy feature of TRSDOS that is lacking in Apple DOS is a flag attached to any file in the directory that has not been backed up.

If you wish to add additional storage, you can use 35-, 40-, or 80-track drives, single- or double-density, with either one or two sides each. Two 80-track, double-sided external drives plus the two internal drives would allow more than 1.7 MB of on-line storage, without resorting to a hard disk. The Apple would need to have 14 disk drives, with seven controller cards (and therefore no printer or serial interface because all expansion slots would be filled), to reach a similar total. Finding files in such an environment could become a nightmare, because Apple DOS only searches a single, designated disk for programs or files.

Apple's DOS 3.3 depends on special utility programs to accomplish the tasks that TRSDOS handles automatically. For example, when you ask Apple DOS for a CATALOG (similar to DIRectory) of any disk, the files on the disk are listed, along with the number of sectors required by each. DOS does not tell you how much space is left on the disk, though. Do you have enough room to save the program you've been working on or a long data file? You can find out by adding up the lengths of all the files already on the disk. There is a special utility that will tell you how much free space is available, but it will wipe out any program you have in memory first.

Apple DOS does not allow password protection for files except by embedding "invisible" control codes in file names. Nor does it automatically add a date stamp to files to allow you to find the latest version of a program easily. One handy feature of TRSDOS that is lacking in Apple DOS is a flag attached to any file in the directory that has not been backed up.

Apple DOS has one obvious bug that is mentioned in its documentation. It is possible, using the RENAME command, to have two or more files on a disk with identical names—what a wonderful way to confuse both the operating system and the person running your programs.

Like TRSDOS, Apple DOS has a VERIFY function. On TRSDOS, VERIFY forces a disk read immediately after each sector of the disk is written to insure that information is stored correctly. In Apple DOS, VERIFY checks file sectors to be sure they are readable. If so, according to the documentation, they "probably" contain the correct information.

Apparently, Apple has decided that most of its buyers are more interested in running applications programs than programming in Basic. Hence its decision not to include a Basic programming manual with its computer. Apple and third-party companies produce many useful applications programs that can compete with any on the market. VisiCalc, for example, was first available for Apple computers. Also, Apple has been more aggressive in trying to capture the educational market, giving a free Apple IIe to any school that asked last summer, and providing excellent implementations of Logo and Pilot. But as a tool for using or learning Basic, the Apple IIe has definite limitations.

Apple users who are interested in programming tend to take one of three possible approaches to overcome the limitations of their machine:

> Most Apple users give up Basic altogether and learn Pascal or Forth, two languages that have been well-implemented for Apple computers.

1. They write special subroutines and programs to perform functions that are normally available in other Basics. For example, you can write subroutines to provide the same functions as PRINT USING, if you wish.
2. They give up Basic altogether and learn Pascal or Forth, two languages that have been well-implemented for Apple computers.
3. They buy a Z-80 card with a CP/M package (about $350) to gain the power of Mbasic which TRS-80 owners already enjoy.

Since I enjoy programming in Basic as well as in a variety of other languages, I am more than satisfied that I made the correct choice when I bought my Model 4. A recent Apple commercial ended with the words, "Someday there will be two kinds of people: those who own a computer and those who own an Apple." Perhaps those words are truer than the ad agency intended; I am happy to own a real computer.

Hardin Brothers teaches high school drama, English, and computer programming. He has published many TRS-80-related articles, and is currently writing a book on methods of extending TRS-80 Basic.

Two Bulletin Board Systems You Can Install

by David D. Busch

Today's bulletin board systems are easy to implement, quick to access, and beneficial to own. The following is a comparison of two BBSs: The Bread Board System and the MCMS Bulletin Board System.

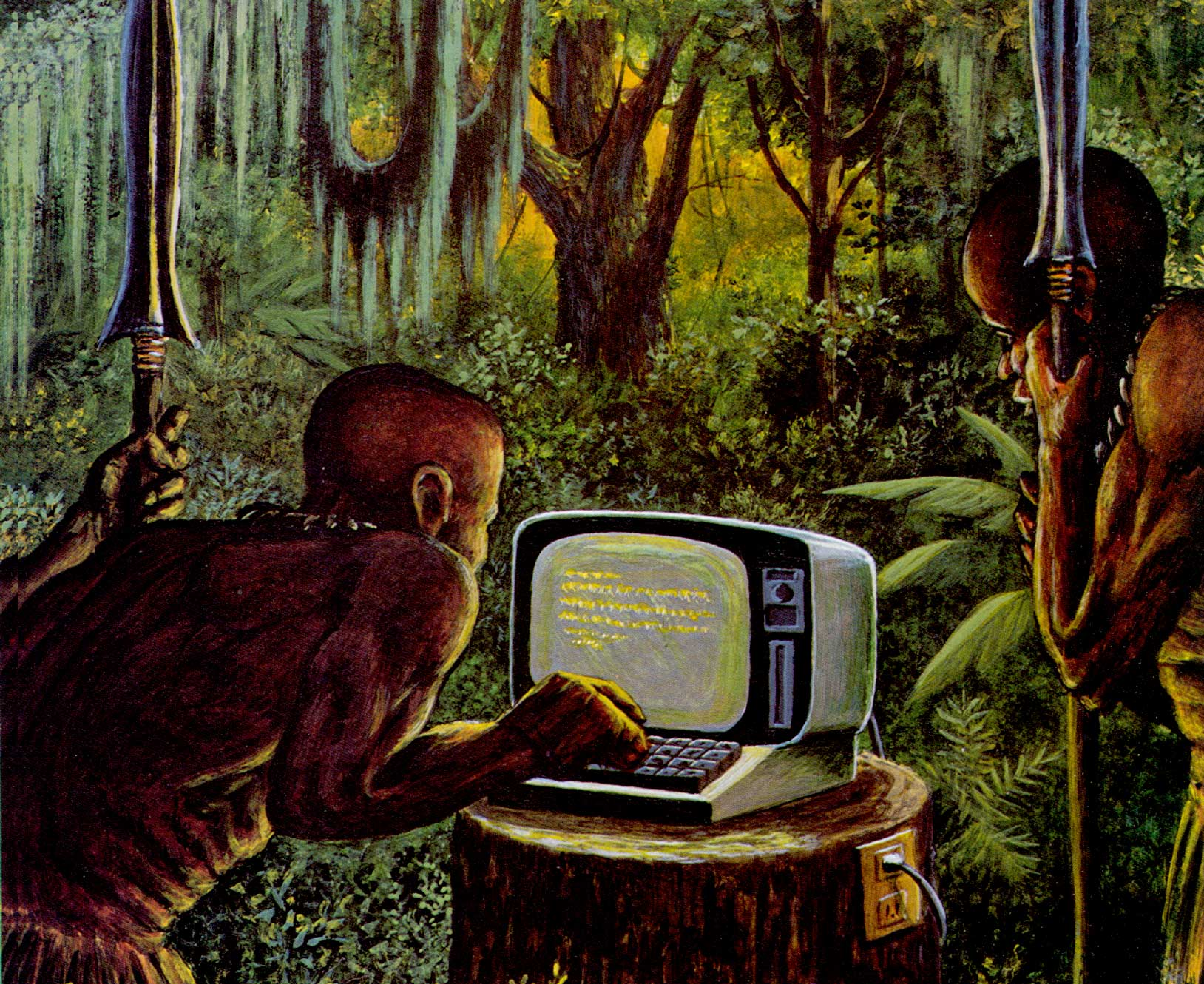
If you haven't checked in with a computerized bulletin board system (BBS) lately, you're probably in for a surprise. Today's BBS is a high-speed electronic communication medium with breathtaking capabilities. Forget about lengthy log-on procedures and waiting for interminable menus to scroll lazily by on your screen. On many systems, the expert user can gain access to the BBS at the press of a key or two, and zip directly to the message board or special interest section desired by touching one or two more.

Several completely different BBSs can also exist on a single computer using a single telephone line, with users of one not aware of the other. Interested in downloading a few public domain Basic programs? The latest boards can do it for you automatically, opening and closing your buffer as required and resending any lines that get garbled along the way.

Those are only some of the changes visible to the user. For the system operator (Sysop), bulletin boards are easier to implement and maintain than ever. The Sysop can access a board from a remote location and perform whatever housekeeping chores are required. In at least one case, it is possible to reconfigure the entire BBS—adding menu items or even entire new message boards—from off-site.

The best part is that you don't need to be a dedicated hobbyist to operate a bulletin board system. Today, anyone with, say, $1,500 for a used TRS-80 Model III and an auto-answer modem can be up and running after about half-an-hour devoted to skimming a manual.

Why would you want to tie up a perfectly good phone line and dedicate a fully-functional computer to BBS duty? If you're a businessperson, you might find a private board the perfect medium



To talk to the natives, you've gotta speak the language...

The world of telecommunications is a jungle of confusing terms. From ASCII to protocols to terminal emulations, there's a lot for the computer novice to remember. That's why we wrote MTERM.

MTERM is the smart terminal package from Micro-Systems Software that goes beyond "user-friendly." MTERM is the first terminal program to be "novice-friendly." You can be an MTERM expert in 30 minutes. And because MTERM functions on so many different microcomputers, there is no need to keep re-learning the program. Universal command compatibiity makes transition easy.

Feature for feature, there isn't a better terminal on the market. Superior programming with top notch documentation make MTERM worth many times its reasonable price.

So in the foreign land of telecommunications, don't despair. Get the best translator money can buy. Maximum power and minimum effort. Finally, these two are in one program.

For your TRS-80®, IBM, Apple®, Zenith, or compatible Microcomputers. Only $79.95 most versions.

TRS-80 and Apple are registered trademarkes of Tandy Corp. and Apple Computers, respectively.

MICRO-SYSTEMS SOFTWARE, INC.

4301-18 Oak Circle, Boca Raton, Florida 33431, Telephone: (305) 983-3390

CIRCLE NO. 31 ON INQUIRY CARD   Orders Only 1-800-327-8724

"The Communications People"

for collecting orders from far-flung, portable-equipped salespeople. A user's group could set up their own BBS to exchange information and messages among members. Computer retailers have long relied on BBS service to their customers as a means of attracting new (and old) visitors to the store or mail order business. Finally, many individuals implement a BBS system of their own because they enjoy the involvement with fellow hobbyists and appreciate the attention bestowed upon even the greenest Sysop.

We looked at two bulletin board system software packages for the TRS-80 Models I and III. Both The Bread Board System (TBBS) from Ebert Personal Computers, Inc., and the MCMS Bulletin Board System from Lakeland Electronics have unique advantages.

From the viewpoint of the caller, both BBSs appear to be very similar. They each allow posting messages and downloading programs, and they have a convenient single-key interrupt feature, which allows the caller to bypass menus by hitting the input key desired at any time while the host is scrolling through the choices. So-called "merchandise" sections can be set up to display a choice of products for ordering.

Each of these bulletin boards also permits the Sysop to specify the authorization level of a user, with various privileges at every successive promotion. If the bulletin board systems of the past were utilitarian Chevy's, the MCMS system is a racy Toyota Supra. TBBS, in a class all by itself, has to rank as a custom-built Rolls Royce, except the operator can decide to have leather upholstery one day and velour the next, depending on his or her mood.

From the Sysop's point of view, MCMS is a very nice BBS. The software will operate on NewDOS 80 for the Model I or Model III, or DOSPlus 3.5 for the Model III only. It is supplied on two disks containing a large number of Basic programs, data files, and other material, with the system already set up for certain default parameters. The defaults include 300-baud operation only, a two-disk TRS-80 system with no external clock attached, and no merchandise section. The supplied system also allows up to 76 users and 85 messages, with first-time callers allowed to download programs.

MCMS could not be simpler to use. To get a basic BBS running, simply make a copy of the master diskette, and type BASIC RUN"MCMS". Assuming that your auto-answer modem is hooked up to your computer and telephone line, that's all there is to it. Of course, your data files and introductory messages will all be dummies (on the order of YOUR NAME GOES HERE), but, those can easily be edited using a word processing program like Scripsit.

From the user's standpoint, MCM is also easy to use, even for the first-time caller. The very clear menu choices can be accessed by pressing a single key. The user does not have to wait for the full menu to be displayed; the system constantly watches the incoming data and interrupts the menu when the caller sends a recognized command. In many menus, the most common reply, or default, is marked with an asterisk. In such cases, just hitting <ENTER> will invoke that choice.

The heart of any BBS is its message system. MCMS' message system is quite sophisticated. Messages can be posted both publicly and privately, and a change in subject is allowed before a message is saved, just in case the user has changed his or her mind. After composing a message, the user can list it using line numbers that can be referenced for editing, or the message can be displayed without line numbers to show what it will look like on the board.

> The operator can set up as simple or as complex a system as he or she desires, with multiple message boards that can be accessed by any users specified. It is even possible to set up entirely separate bulletin boards.

Have you ever been halfway through a reply to a question, only to discover that you can't remember exactly what the question was? MCMS allows the responder to temporarily save a message and go to any other part of the board. Then, typing a special command (L from the main menu, or R from the end of message prompts) will restore the message you were working on. This is a very useful feature.

MCMS can be modified to a certain extent. The 11-page documentation provides a list of variables used by the core Basic program. The definitions can be modified by the operator as desired. For example, SYSOP$ equals the Sysop's first name; NA$ is the system name. TL equals the time limit for callers, while X1 is the default drive number for system data files. X7 is the default drive number for upload files.

Callers can be assigned one of nine levels of "authorization." This is used chiefly by "remote" Sysops who may need to access some of the files through another computer. The level nine Sysop has access to all files; the level eight user can only access user log records. Both must supply the BBS's master password at log-on.

Various utilities are supplied with MCMS, including user log and listing utilities, a file compare program, and a special program for the Hayes Smartmodem.

The Bread Board System (TBBS) is more than a bulletin board. It's more akin to a bulletin board "construction set." The operator can set up as simple or as complex a system as he or she desires, with multiple message boards that can be accessed by any users or group of users specified. It is even possible to set up entirely separate bulletin boards—with their own Sysops and menus —that are invisible to all the other users.

The basic TBBS board looks much like MCMS. It is a fast bulletin board system that allows single-key entry of commands and displays helpful menus. Uploading and downloading of files is supported, along with a merchandise section for the commercially-operated boards.

TBBS allows the Sysop to define each of the message boards and features. Let's look at a typical message board and the options that can be defined:

1. Menu. The menu listing can be set up to the Sysop's specifications. In this case, we have highlighted the first letter in the menu option to indicate that this message board can be accessed by typing in that letter, "M":

```
*** TBBS MENU ***
<M>essages
```

2. Privilege. TBBS allows the Sysop to set each user's privilege level from 0-255, with the higher numbers according more privileges. This level can be set at any desired level, and also can be assigned as a default upon first log-on.

For example, TBBS can be set to assign a privilege level of 25 to any first-time caller. That user would retain that level until the Sysop makes a change through the user-log editing program. Say a troublemaker has been leaving obscene messages. The Sysop can reduce that user's privilege level to 0, which

would be lower, even, than a first-time caller. If the troublemaker catches on, of course, he or she can always log on again with a new name to gain a 25 privilege rating once again.

The Sysop can, at his or her option, set the privilege level of any board so that only those with that level or higher can see it on the menu or access it. To those with lower authorization, the board is invisible.

3. AUTH 1 and AUTH 2. TBBS really gains flexibility with these two additional layers of authorization. Each consists of a flag-string of eight characters —either hyphens or X's. Every user has their own AUTH 1 and AUTH 2 flags, as well. A given board can be marked with appropriate flags, and *only* those users with matching X's may use that board (or other feature).

Say our message board is intended for the private use of TRS-80 Model 4 owners who have paid a $10 registration fee. That board might be given AUTH 1 and AUTH 2 flags such as AUTH 1 = --X-----, and AUTH 2 = ----X---. The user would have to have an appropriate privilege level, plus matching X's in their respective AUTH 1 and AUTH 2 flags in order to access that board. The Sysop, who will be able to access any board in the system, will have a privilege level of 255, and AUTH 1 and AUTH 2 codes of XXXXXXXX.

4. Key. This is the key hit by the caller to access that menu choice. For an <M>essage option, it might be M. Any other character, such as an exclamation point, can also be used.

5. Type. Some 45 menu type options may be selected to specify exactly what type of function should be carried out when that menu item is chosen and where control should return (e.g., back to that menu, or to some other).

6. Optional data—such as the name of the message board. Because of the complex possibilities of the various authorization and privilege levels, you can see that it is simple to set up a whole hierarchy—or several hierarchies —of message boards accessible to many different groups of users. Menu choices don't appear to unauthorized users. (Actually, they can appear, if desired, but access can still be limited by AUTH flags.) It is even possible to assign outsiders as Sysops of their own special interest groups.

Remote operation of TBBS is among the most flexible I've seen. By logging on and supplying an appropriate password, the Sysop can access the editing modules, if desired, and redefine menus or message boards from a remote location. It is even possible to go to DOS, copy files, kill files, and perform other tasks from a remote location. Because TBBS remains active in protected memory, the Sysop can return control and log off at any time. The effect is that of being able to operate your computer from across the country, if necessary.

Nearly every feature of TBBS can be defined by the operator. Passwords can be required or not. All of the messages and menus can be altered as desired. Message boards can be public, offer private messages, or allow EMail-only-type communications. The programs included even have a "survey" module that allows a poll of members to be taken, or question-and-answer sessions to be held.

All of this is complex and takes some time to understand, but TBBS is set up

```
TBBS Top Level Menu

<R>ead . . . Read Msgs on TBBS board
<Q>uick-scan Msgs on TBBS board
<L>eave . . . Leave Msg on TBBS board
<K>ill . . . Kill Msg on TBBS board
<M>ail . . . Electronic Mail Section
<G>oodbye . . . Terminate Session
<B>ulletins
<#>Userlog . . . List of callers
<X>pert . . . Change user level
<E>ditors . . . Remote SYSOP section

Command: M
```

```
<R>ead your Mail
<S>can your mail
<L>eave Mail for other users
<K>ill your mail
<E>xit back to TBBS
<X>pert . . . Change User Level
<G>oodbye . . . Terminate Session
<T>ime on System

Which One? R
```

```
Type P to pause, S to stop, N to skip to
next msg

Read Electronic Mail:

<T>o you
<F>rom you
<B>oth To and From you
<M>arked Mail
<A>bort Reading Mail

Which One? B

No mail found.
```

Sample TBBS conversation

How to Submit Articles

Interface Age is always happy to review unsolicited articles or query letters to consider for publication. Our primary focus is on microcomputer applications for businesses. The fields of law, medicine, retail sales, accounting and record keeping, filing and all manner of various office functions are emphasized. Case studies of how particular businesses computerized their operations are especially encouraged. Step-by-step procedures, including planning, product selection and implementation should be detailed.

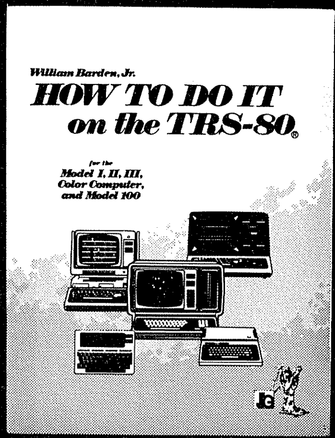
Good quality program listings, accompanied by articles explaining the programs, are also sure bets for serious consideration. The listings should be no more than 60 characters wide, with no wrap-around lines. Unlined paper and a new ribbon should be used. Sample runs should also be included. In the article, variables should be described. The system utilized in composing the program should be detailed—operating systems, language type and version, and any necessary peripherals.

Submittals should be prefaced by a brief synopsis of the article. Manuscripts should be typed or printed out double-spaced with one-inch margins. Minimum text length is eight pages, whether or not the article is accompanied by a program listing. Photos should be numbered and have a brief description attached to each. Tables, listings, etc. should be on separate pages and each should have a caption. Authors are requested to submit a statement of background and expertise.

The publisher assumes no responsibility for artwork, photos or manuscripts. No acknowledgement is made unless the submission is accompanied by a large, stamped return envelope. A minimum of six weeks should be allowed for a response; it is requested that authors do not phone for information about submittals.

The submittals should be addressed to: Editorial Department, *Interface Age* Magazine, 16704 Marquardt Ave., Cerritos, CA 90701.

the other half of the TRS-80: THE MIND!



THE BOOK YOU CAN BOTH READ AND UNDERSTAND

by William Barden, Jr.

IJG

CompuThoughts

I can't begin to tell you everything about this book— it's unusual. The first things you notice about it are that it has no page numbers and the "table of contents" has over 2,000 entries in alphabetical order. Here's why:

We use TRS-80s— from accounting to typesetting. The variations of BASIC, applications software, hardware and operating systems is more than can be remembered. Nothing gets done when your nose is in a manual.

Why can't "they" make a book that has everything in one place? Hell's fire, "we" are "they!" I told Bill Barden that if he would write such a book I would do two things: 1) I'd publish it and 2) I would quit bitching.

I use it everyday. It answers my questions. It'll answer yours too.

H.C. Pennington, President

IJG books are sold at booksellers and computer dealers world wide. If your dealer is out of stock, order direct. Include $4.00 for shipping and handling. Foreign residents include $11.00 in U.S. funds for shipping and handling. IJG, Inc., 1953 West 11th St. Upland, California 91786. (714) 946-5805

IJG

The Thoughtware Company™

CIRCLE NO. 22 ON INQUIRY CARD

> Both of the systems recovered from errors quite well.

for the non-computerist. All editing and operational functions are controlled by menus. Once you understand what the menus will do, the system literally configures itself for you.

Ebert Personal Computers even supplies DO/BUILD files for various operating systems that will set up a typical system for you. From opening the package to getting on-line takes half-an-hour or less.

Both TBBS and MCMS are easy to get started. MCMS comes already configured for a basic BBS system, and tells you how to make the modifications you want. TBBS has a mind-boggling number of options, but can use job control language to set up a simple BBS automatically. Once you play with either for awhile, deciding on custom modification is bit easier.

Each of the bulletin board systems is about equally friendly to new users, while allowing veteran callers to by-pass menus quickly. From the Sysop's point of view, MCMS requires a bit more computer knowledge to modify and understand. TBBS can be customized by anyone with the patience to absorb all the information at hand.

MCMS and TBBS are both outstanding boards at the simplest level. Many people may not need the extreme flexibility in setting up sub-boards, as provided by TBBS, but then it is more of a system than a simple host program. I'd compare TBBS more with Lotus 1-2-3 or VisiCalc than I would with other BBS programs.

Both of the systems recovered from errors—such as disconnection—quite well. Other error traps allowed limiting the time of each call. TBBS, because of its complexity, can be configured in a ridiculous manner by mistake. A message board can be set up that can only be accessed by the Sysop, for example. You do have to know what you are doing to avoid all problems with either program.

Lakeland Electronics provides a number you can call with questions. Ebert has two bulletin boards available, one for Sysops, and one for the general public. The company does ask each registered Sysop to call in at least once a month to get news of updates and improvements.

The MCMS documentation was somewhat sparse and left a few questions. At 100 pages, the TBBS manual was much more thorough, and covered a lot more ground. I found, though, that most operators would have everything they need to know from either manual. The TBBS documentation even includes data to help Assembly language programmers make modifications to the program code.

If you want nothing more than a state-of-the-art BBS, consider MCMS. If you need to set up a BBS to your own detailed specifications, then TBBS is the only choice.

David D. Busch has written more than 200 articles on computers, applications, and word processing. In the past two years, he has devoted special attention to personal and small business computers as a programmer, reporter, and observer.

Product:
The Bread Board System

Manufacturer:
Ebert Personal Computers, Inc.
4122 S. Parker Road
Aurora, CO 80014
(303) 693-8400

System Requirements:
TRS-80 Model I, III, or 4
Printer (optional)

Special Requirements:
two disk drives
NewDOS, LDOS, or DOSPlus.

Price: $199.95

Product:
MCMS Bulletin Board System

Manufacturer:
Lakeland Electronics
401 W. Hawthorne
Round Lake, IL 60073
(312) 546-5671

System Requirements:
TRS-80 Model I or III—48 KB
two disk drives

Special Requirements:
NewDOS or DOSPlus 3.5
Model I disks must be double-density
additional drives (optional)
printer (optional)

Price: $49.95

DiskCount Data™

WHERE QUALITY PROGRAMS MEET COMPETITIVE PRICES

2701-C W. 15th • SUITE 324 • PLANO, TX 75075 • (214) 680-8268

SUPER BOWL SAVINGS

All MOD III programs specified in this ad will run on the Model 4 (in the MOD III mode).

SUPER UTILITY PLUS VERSION 3.2
By Powersoft

SU+ 3.2	79.95
Inside SU+	19.95
Retail	99.90

SALE . . . 69.95

30% SAVINGS

80 micro readers' choice 1982

NEW SUPER UTILITY+ VERSION 3.2
Includes Operators Manual and 2nd Backup Disk
The Book Inside SU+ 3.0 Included FREE
A 19.95 Value at no extra charge
Experience a legend with the program voted as the outstanding utility of 1982.

MOD I or MOD III — Protected Media

INFOSCAN
By The Small Systems Center

Retail 49.95
Special 44.95

NOW 44.95

InfoScan is an information utility that is used to organize store and retrieve any type of useful information in any format

It combines the features of a word processor data base manager and information retrieval and display utility You compose information records on the screen,using the word processing feature in any format you like Each record is described by 1-17 character key phrase that you specify Once created the information file s keyphrases can be scanned as a list on the screen, using an impressive key scan bracket You scroll the keyphrases thru the brackets with the arrow keys on your keyboard When the desired Keyphrase is found, you press a single Key and the corresponding information record is retrieved from disk and displayed on the screen The record displays in exactly the same format as you used to create it You can even design your own forms to "fill in"

A stunning visual presentation it utilizes the display window technique Menus, Keyphrase lists, displayed text, and status information are displayed in a series of display windows on your screen Impressive graphic display panels are used to enhance the visual orientation of the product and your information

MOD I/III MINIMUM 48K 1 Drive

MULTIDOS
BY Cosmopolitan Electronics

Reads Other Operating Systems

VERSION 1.6 $99.95 Now 84.95

The latest release of an outstanding operating system or

Buy VERSION 1.5 MOD I or **VERSION 1.3 MOD III**

at an unbelievable

$49.95

MOD I or III Specify

THE TOOLBOX for LDOS
by Powersoft

Previously 99.95
Reduced to 69.95

SPECIAL 59.50 SALE

SAVE 15%

PCHECK/CMD	PCOMPARE/CMD
PFIX/CMD	PCLEAR/CMD
PREFORM/CMD	PSS/CMD
PVU/CMD	PMAP/CMD
PERASE/CMD	PMX/FLT MX80
PMOVE/CMD	PHELP/CMD
PDIRT/CMD	PBOOT/CMD
PASSGO/CMD	PFILT/FLT
PUN/CMD	DVORAK/FLT
PEX/CMD	DVORAK/JCL
PMOD/CMD	CODE/JCL
PFIND/CMD	DECODE/JCL

MOD I or MOD III

POWERDOT II
by Powersoft

NEW VERSION
RETAIL $59.95

The ULTIMATE in graphics design. This version is MUCH MORE POWERFUL than previous versions and includes BETTER documentation as well. New features include AUTODRAW™ and CIRCLE commands. Now allows you to design your own character sets! Includes lots of examples on disk. Your screen is only a "picture window" to a much larger drawing area! You are only limited by disk storage, not memory! Works on EPSON Series (Graftrax of Graftrax+ required) or the C.Itoh 8510 (PROWRITER or NEC version). Now available for DMP-2100, DMP400 and LP-8. SPECIFY PRINTER

SAVE OVER 15% NOW 49.95

LC COMPILER AND EDAS
By Misosys

LC Compiler Retail	$100
EDAS Retail	$100
SPECIAL BUY BOTH FOR . . .	$139.99

SAVE OVER $60

Now program in the "C" language compiler subset for LDOS. LC generates Z-80 EDAS IV source code as output. Compiled programs run on both MOD I and MOD III. Save $60.00 with the best LDOS editor assembler package on the market. Over 200 pages of documentation. Requires 2 drives and 48K.

FOR LDOS MOD I/III

THE TEST GENERATOR
By Big G Software

ONLY 34.95

ATTENTION EDUCATORS AND STUDENTS

STUDENTS
This program is an ideal study aid for home computer use. A student types in review questions, vocabulary words, notes, etc. and the computer asks the questions. The student has two chances to answer correctly. The the computer gives a score and shows questions answered incorrectly. The student then has the option to recycle incorrect answers.

TEACHERS
Type in up to 100 questions and answers to compose a test. The test may be printed in Multiple Choice, Matching, True-False, or Short Answer format. Questions may be printed in order or at random. Answers may be printed as questions, and questions as answers. An answer key is automatically printed for each test. Tests may be duplicated using Xerox, spirit masters, etc.

MOD III

SuperDirectory
By Computer Shack

Manufacturer's Suggested List	49.95
DiskCount Data Special	44.95

SAVE 10%

- **Build a Complete Library of All Your Programs**
- **Automatic Density Recognition**
- **Automatic Track Count Recognition**
- **Automatic DOS Recognition**
- **The Best Directory On the Market**

READS ALL DOS SYSTEMS

A great new directory program with its own operating system written by Vernon Hestor. This program will read any DOS there is, even compatability between MOD I and MOD III with doubler installed. The MOD I program can read most MOD III disks and the MOD III program can read most MOD I disks. 80 track double density is also supported

---

MACRO-MON THE SHADOW
by Jake Commander

BELOW DEALER COST

A HUGE DISKCOUNT DATA BUY PASSES THESE UNBELIEVABLE SAVINGS ON TO YOU

Formerly Manufactured by

ADVANCED OPERATING SYSTEMS

Now a DiskCount Data exclusive

The industry's standard and the finest break point disassembler and machine language monitor available. The package includes a rich, padded easel binder and complete 2 color documentation. Now single step through your CMD files, view & change assembly op-codes, memory, etc. Used by Kim Watt for the editing of SuperUtility+

MOD III Disk	69.95	Now Only	34.99
MOD I Disk	64.95	Now Only	32.99
MOD I Tape	59.95	Now Only	29.99

Quantities Limited & Subject to Prior Sale

LYNN VIDEO INSTRUCTION TAPES
VHS OR BETA FORMAT

NOW ONLY $36.99*

- Profile III+
- Super Scripsit
- Lazy Writer
- Mod I/L-II Basic
- NewDos 80
- Mod III Basic
- Scripsit
- TRSDOS 1.3
- TRSDOS 6.0
- Ldos
- Visicalc
- DosPlus
- MultiDos
- Trsdos 1.3 disk I/O
- Trsdos 6.0 disk I/O

SAVE

*MOD III
*MOD I
BASIC
46.99

GAMEPAK 3
by Micro Mining

ONLY 29.95

3 GAMES FOR THE PRICE OF ONE

This package by James Talley of Kid-Venture fame includes 3 games on the same disk. Funny Face is a program that draws over 1 million different faces on the screen as you select the different facial features. You can play "line-up" where you select the face from up to 6 like faces or have the computer draw faces in the automatic mode. Matchmaker is a program like the TV game "Concentration" where you select identical graphics matches from a grid. It keeps a running point score. Repeat After Me emulates the famous "Simon" game where you have to enter a repeat sequence of numbers to match the computer. Sound included. Funny Face alone is well worth the price. Kids will adore this 3 game selection

MOD I/III

DOSPLUS
by MicroSystems Software

SALE

SPECIALS ON ALL VERSIONS

VERSION	RETAIL	SALE
4 MOD 4	$150	$119
3.5 MOD I or III	$150	$119
3.4 MOD I	$150	$59.95
3.3 MOD I SD OR DD	$100	$44.95

NOTE: We have several MOD III 3.5's with the Hard Disk Drivers included at $139.95.

POWERDRIVER $29.95
By Powersoft

NEW PRINTER DRIVERS

This new generation of custom printer drivers allows you to utilize all of SUPERSCRIPSIT'S features with your EPSON (Graftrax Required), PROWRITER and C.ITOH F-10 STARWRITER printers. You can now utilize all of the printers custom features such as compressed, expanded print, underlining, bold-face, super-scripting and subscripting. All drivers can be called from within SUPERSCRIPSIT at "document open time" Order by printer designation as follows:

POWERDRIVER E	EPSON MX70/80/100
POWERDRIVER P	C.ITOH PROWRITER
POWERDRIVER S	C.ITOH STARWRITER

MOD I or MOD III

ULTRATERM
by United Software

OVER 15% OFF

ULTRATERM NOW 49.95
Version 2.0 w/Auto Logon 67.95

This communications package is one of the most versatile available. It includes a full featured intelligent terminal program, host program, direct file transfer function and hex conversion utilities for bulletin board downloading plus • Support for virtually all manual and auto dial modems • Exclusive Ultra Term direct disk file transfer function allows full control from the transmitting computer and unattended operation at the receiving computer • Split screen feature lets you type outgoing messages on the bottom half of the screen while incoming messages are displayed on the top of the screen • Full Line printer support • Universal ASCII format file transmission and reception with memory input output buffer supports delays between characters, delays between lines, and prompt controlled transmission for compatability with virtually all systems. A counter continuously displays the amount of free memory remaining whenever the buffer is in use • Full featured host program that supports PRINT and PRINT [illegible]

THE HOME ACCOUNTANT™
By Continental Software

ONLY 69.95

An Outstanding Financial Planner

• Maintains up to 100 budget categories • Keeps track of up to five checkbooks • Prints checks if desired • Prints a personal balance statement, income and expense summary • Prints net worth statement • Provides fast bank reconciliation • Allows file extensions on multiple diskettes • One program handles cash, checkbooks, credit cards and other liabilities and expenses • Unlimited annual transactions, fiscal or calendar year • Transactions may be "split" among different budget categories • Flags transactions for tax purposes • Maintains transaction history • Provides Hi-Res graphics for any category by bar graph

"The program itself does just about everything you'd ask of a personal finance package." Popular Computing, November 1982

MOD III

POWERMAIL PLUS
By Powersoft

Retail 150 00
Special 112 50

SAVE 25%

Features Over the Original PowerMAIL

Separates any category for you in any manner you wish. This new version can span drives and may use multiple floppies. The program will sort on any eight levels if you wish. Separates your flags and puts them into another file. Print Options Labels & Listing. One choice is the simple "telephone book" type listing. Two of the other modes give you not only the listing of names but also either FLAG numbers, or the actual FLAG descriptions. The final print mode is what we call THE SALESMAN'S NOTEBOOK™. It prints out the whole list or whatever parts you specify with all info including full flag descriptions in each record. Then the word "Notes" is printed with several lines of blank space for making those notes in the field'

MOD I/III/4 2/12/16
SPECIFY

CP/M 2.2

TRS-80 MODEL 4
Only 199 95

Now, run CP/M on your Model 4 and unleash the powerful features resident in your computer.

● Includes INTERCHANGE™, a utility that allows reading, writing and copying 20 different manufacturers' disk formats such as IBM, KAYPRO, OSBORNE, XEROX, etc ● Includes MEMLINK™, a unique feature that uses the optional 64K RAM memory as a fast disk drive. ● Complete with all these CP/M utilities: ASM, DDT DUMP, ED, LOAD, PIP, STAT and SYSGEN. ● Includes MODEM 7, a powerful public domain communications program ● Supports 80 x 24 video, reverse video, direct cursor addressing and more ● Utilizes the Model 4 function keys and allows user defined keys. ● FORMAT utility permits up to 52 disk formats to be constructed, all menu driven. ● Ready to run in the standard 64K Model 4. The additional extra cost 64K RAM upgrade not required.

MISOSYS

Software for the Model 4

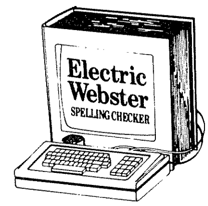
PRO-CESS	37.95
PRO-CREATE	94.95
PRO-CURE	46.95
PRO-DUCE	37.95
PRO-LC	139.95

SEE MISOSYS AD FOR DETAILS

ELECTRIC WEBSTER
by Cornucopia Software

149.95 SAVE 20% WITH CORRECTING FEATURE

ONLY 119.95



SYSTEM DIAGNOSTIC
By Howe Software

ONLY $99.95

Complete diagnostic tests for components of your TRS-80. Separate tests for ● ROM ● RAM ● VIDEO DISPLAY ● KEYBOARD ● LINE PRINTER ● CASSETTE RECORDER ● RS-232-C INTERFACE ● DISK DRIVES ● DISK CONTROLLER ● INDIVIDUAL TESTS AND CONTINUOUS SYSTEM TESTS

MOD I or MOD III

6.0 PLUS
by MicroSystems Software

Enhance Your Model 4 TRSDOS 6.0 and Disk Basic

39.95

A great value from the DOSPLUS folks, these utilities for your MOD 4 DOS add some really important features.

Enhancements include:

DISKZAP	MAP	REF
DISKDUMP	RESTORE	RESOLVE
DIRCHECK	SORT	BE1
	SR	RF2

MOD 4

EYE EASE GREEN CRT SCREEN FOR

TRS-80 MODEL III

A MUST FOR THE SERIOUS COMPUTERIST

MACHINE BEVELED EDGES
READY TO INSTALL
SELF BONDING
DUSTPROOF SEAL
OPTICALLY CORRECT
NO DISTORTION
REDUCES EYE FATIGUE
SHATTERPROOF
ENHANCES IMAGE

SAVE OVER 30%
MOD I, II, III/4
NOW 16.95

$24.95

GREEN WINDOW

PREFERRED BY COMPUTER PROFESSIONALS EVERYWHERE

OPTICOM

ENIGMA
by Powersoft

ONLY 39.95

This encryption program is based on the ENIGMA cypher machine used by the Germans in the Second World War. It incorporates the improvements which, by the admission of Allied Intelligence who participated in cracking the Enigma code, would have made the system almost completely impenetrable! A 29-character key is used for encrypting textual material, yielding over TRILLIONS of possible keys and encoding combinations. A technical description of the German ENIGMA machine and how it is simulated (and improved upon) is given in the manual. The perfect program for students of cryptography, business persons who want to transmit sensitive data over open data links (voice or modem) or TELEX lines, and anyone else who wishes to maintain confidential information.

MOD I/III

PowerDRAW
by Powersoft

Retail 39.95
Special . . . 34.95

SAVE OVER 10%

A full screen graphics editor, POWERDRAW is 100% assembly language. You can create screens of graphics, save to disk, merge them, run in sequence like a movie, merge text with graphics, and write your own game or business application screens! PowerDRAW saves graphics to disk or tape so that they can be recalled at a later time in the following formats

0) CONDENSED TOKENS	4) BASIC Strings
1) EDAS Source file format	5) BASIC arrays (New in version 1 3!)
2) EDTASM Source file format	6) Load File Format
3) BASIC data statements	

THE TOOLBELT MOD 4
by Powersoft

Powersoft does it again with a collection of 16 of the most popular LDOS TOOLBOX utilities re-published for TRSDOS 6.0 running on the MODEL 4. Now enhance your MOD 4 TRSDOS with this powerful set of utilities. Complete with detailed manual on how to use each utility program to do the many things the TOOLBOX for LDOS was famous for.

49.95

Model 4

GRAPHIT
by Powersoft

ONLY 39.95

A simple, easy-to-use program for portraying graphs on a microcomputer. Creates and uses files in Visicalc DIF format, so it will be able to graph Visicalc data as well! Graphs may be changed at the touch of a key; graph scales can be compressed or expanded. Display and print just one portion of the graph or all of it at once. Extremely user-friendly and contains multiple HELP screens. GRAPHIT will print the raw data and produce graphs on any lineprinter. No special lineprinter options required.

MOD I/III

EPSON™ Driver Compiler
by Powersoft

29.95

This is a program which allows you to construct your own printer character table for an EPSON bit-graphics printer, and compile a relocating printer driver which will produce those characters! Character tables can be saved to disk and manipulated in a variety of ways (including rotation of each character 90, 180 or 270 degrees) before the actual compile process. Several example files are included on the disk. This program requires an EPSON MX-80 with GRAFTRAX or GRAFTRAX-Plus, or an FX-80

Mod I/III

BBS-80
by MicroSystems Software

ONLY 74.95

Your computer becomes a mini-communications network. BBS-80 allows a person to exchange messages or information via the telephone.
Access can be completely public, private to authorized users or limited to certain individuals.
BBS-80 can leave messages on another terminal acting as a rapid, electronic mailman.
BBS-80 provides some of the most advanced and most convenient features ever incorporated in a microcomputer bulletin board system including:
Users may send private mesages that may be read only by the addressee.
Multi-level System security allows up to 15 levels of security in both the message base and the upload/download database.
BBS-80 "remembers" each regular user and stores terminal information, security access levels, last message retrieved during last call etc.
Individual user passwords.
Users may "upload" or transmit data and programs files to BBS-80.
These programs may then be "downloaded" to other BBS-80 users.
BBS-80 is self maintaining. Message space is automatically reclaimed when messages are deleted.

MOD I or III

IMPAKT
by Powersoft

39.95

An outstanding value for Basic Programmers, IMPAKT is a relocatable machine language module that requires no modification to basic but will do the following: search & replace ● remove remarks ● remove spaces ● remove colons ● remove goto following then ● compress or decompress ● renumber ● trace by line # or step without destroying screen ● set break points in trace mode ● join lines ● copy lines with auto renum ● move blocks and check for validity ● display memory useage stats ● decode packed string graphics ● conv upper case to lower case strings ● auto save & load from disk and many more

MOD I/III

MORE DISKCOUNTS

PROGRAM NAME	SPECIAL
BASIC S COMPILER SYSTEM I/III	39.95
MICROTERM I, III or 4	69.95
SCRIPLUS 3.0 I/III	29.95
DOSPLUS II MOD II/12 or Z-80 MOD 16	199.95
M-ZAL EDITOR ASSEMBLER RELEASE 3 MOD I or III	84.95
ZUES EDITOR ASSEMBLER MOD I/III/4	74.95
MASTER MECHANICS SET FOR LDOS (9 UTILITIES) I/III	34.99
LOAN AMORTIZATION (FULL AMORT SCHEDULE) III	29.95
E-BASIC FOR MULTIDOS MOD I OR III	49.95
SUPER UTILITY TECH MANUAL 3.(X)	11.95
INSIDER SUPER UTILITY 3.(X)	16.95
INSIDE SUPER UTILITY 2.2	14.95
THE CUSTOM TRS-80 & OTHER MYSTERIES	26.95
HOW TO DO IT ON THE TRS-80 & OTHER MYSTERIES	26.95
TRSDOS 2.3 DECODED & OTHER MYSTERIES	26.95
MICROSOFT BASIC DECODED & OTHER MYSTERIES	26.95
BASIC FASTER & BETTER & OTHER MYSTERIES	26.95
MACHINE LANGUAGE DISK I/O & OTHER MYSTERIES	26.95
BASIC DISK I/O & OTHER MYSTERIES	26.95
TRS-80 DISK AND OTHER MYSTERIES	19.95
DISKETTES 5¼ IN. SSDD, IN PLASTIC EASLE BOX	24.95
LDOS 5.1 MOD I OR III	119.95
PROFILE 3+ COMMANDS (17x22 WALL CHART)	*4.00
VISICALC COMMANDS (17x22 WALL CHART)	*4.00
SUPERSCRIPSIT COMMANDS (17x22 WALL CHART)	*4.00
MODEL III BASIC COMMANDS (17x22 WALL CHART)	*4.00

CIRCLE NO. 12 ON INQUIRY CARD

*NOT SHIPPED SEPARATELY

FREE
OFFER

Any order of over $100 or more from this ad will receive one choice of the following absolutely **FREE**

Green Window I/II/III/4,
SU+3 (X) Tech Manual
Macro-Mon The Shadow

Any order over $200 will receive one choice of the following FREE
Multidos MOD I 1 5, MOD III 1 3 or GAMEPAK 3

DiskCount Data ™

214-680-8268

Send Cash, Check or Money Order
Please add $3 00 for postage and handling
additional $2 00 for C O D's

●

Foreign orders welcome, please specify air or surface. All shipping charges assumed by purchaser

●

When ordering by mail, please specify computer model number (I, II, III or 4), drive configuration, and memory size

Phone Your Order In Today Or Mail To
DISKCOUNT DATA
2701-C W 15th St, Suite 324
Plano, TX 75075
Office Hours Mon-Fri 10A M to 9P M C S T

MasterCard AND VISA Cheerfully Accepted

The CAI Journey

For one professor in Menomonie, WI, Computer Assisted Instruction has raised productivity 33 percent, has improved efficiency, and has given students a taste of what computers can do.

by Arthur C. Matthews

How would your administrator react if you came to him or her and said, "Boss, how would you like me to improve my ability to teach my students, free up a lot of time for individualized instruction, improve the efficiency of myself and the students, give the learners a taste of what a computer can do. . . and, oh yes, raise my productivity 33 percent and pay for my time and the computer's in one school year? Would your boss faint? Cry Hallelujah? Pat you on the back?

My CAI (Computer Assisted Instruction) journey didn't begin quite that way. In the administrative blackboard jungle, there are forms to be filled out, feasibility studies to be made, and politicking to be accomplished.

I teach radio production at the UW-Stout in Menomonie, WI. Even though we are a technically oriented school, computers remain a mystery for many teachers. I got "computer religion" on my own. I saw the light in September 1982; now I'm using two CAI multi-media work stations.

I started by becoming a magazine-aholic, reading every computer related magazine I could find about what people were doing with, for, and to these mindless brains. I ended up choosing the TRS-80 Model III (I and 4 also work) as my computer because:

1. The software I wanted was already developed by Tandy.

2. I couldn't use color—text appeared to be too fuzzy and the graphics not detailed enough for my purpose.

3. I demanded a reliable machine. I didn't want to become the unpaid field tester for an ambitious manufacturer.

4. I required good repair facilities without having to pay for a service contract every year.

5. I wanted a full keyboard (rather than membrane or chiclet) to encourage touch typing.

Radio production deals with sound and equipment. My multi-media workstation plan integrates illustrations and stereo sound. My graphics solution lay in a slide projector with a short range viewing screen. For stereo sound, I chose the Realistic playback-only stereo cassette. I added a small amplifier with four pairs of volume adjustable headphones and the Model III. I had my multi-media workstation ready, but what did I intend to do with the ungainly beast?

Radio Production is a laboratory course. The students spend two hours a week in lecture (one mass section) and two hours a week in lab (10 sections). At the time I began my CAI safari, I had three workstations in the old lab: an on-the-air studio, Production Studio #1, and Production Studio #2.

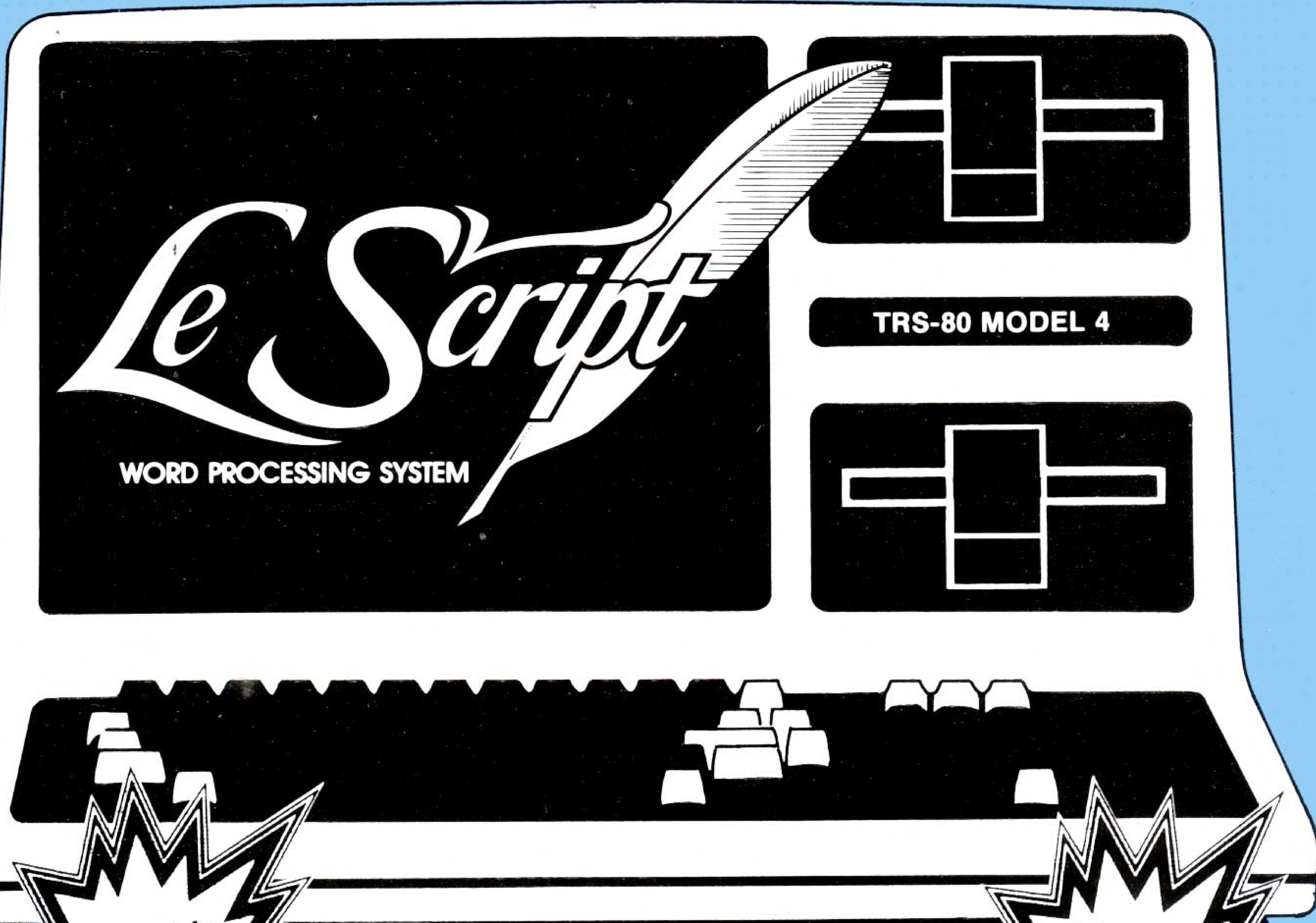
By adding a multi-media workstation incorporating the TRS-80 Model III, I was able to up the number of students in each lab section from 12 to 16 (a 33 percent productivity increase) and still meet my other goals.

In the old course—1982 BC (Before Computer)—I had a unit called "The Knowledge Project." The students could learn about audio systems or take a series of tests I prepared on radio station operation. If the student passed the tests, he or she could apply for their FCC (Federal Communications Commission) Form 753, which allows the holder to be a DJ on the student-managed portion of WVSS.

In September 1983 AD (After Development) the students had five "Knowledge

See us at
SOFTCON
La Superdome
New Orleans
Tuesday-Thursday
February 21-23, 1984
BOOTH P-2063

TRS-80 MODEL 4
Word Processing Program



Available NOW!

ONLY 129^{95}

80 x 24 DISPLAY

- Works with **TRSDOS 6** or any **Model III DOS.**
- **90K TEXT BUFFER** when expanded memory is installed.
- Built-in **FORM LETTER** and **MAIL-MERGE** capabilities.
- Performs **PROPORTIONAL-SPACE** right-margin justification on over 60 different printers (all drivers included **FREE**).
- Same powerful editing features as **ZORLOF II,** with dozens more features added.
- Le Script also available to run on **MODEL I, MODEL III, LNW-80, PMC-80,** and **MAX-80** (80x24 display).

FREE

- 30-Day Support to Registered Owners
- Subscription to LeScript Newsletter

LeScript costs $200 less than competing products matching its capabilities.

VISA MasterCard

CALL 1-305-259-9397

ANITEK

ANITEK SOFTWARE PRODUCTS
P.O. BOX 361136
MELBOURNE, FL 32936

FREE SHIPPING WITHIN THE U.S.; OUTSIDE THE U.S. ADD $10.00 FOR SHIPPING; FLORIDA RESIDENTS ADD 5% SALES TAX. ALL ORDERS PREPAID BY CHECK, MONEY ORDER, CREDIT CARD OR C.O.D.

CIRCLE NO. 5 ON INQUIRY CARD

Project" choices:

1. Learning about Computer Assisted Instruction (CAI),

2. Viewing a six-part introductory Audio course,

3. Taking a Recording Techniques course,

4. Interacting with a computer to prepare for the Form 753 test, or

5. Writing a paper on some phase of audio.

I use the computer to present information, to encourage self/other evaluation, to increase computer literacy, and to introduce new equipment.

The students' first crack at the computer comes when their group is scheduled for "KN/PROJ." The goals of this project are beginning computer literacy and information. There are two modules: "How to Work With a Computer" (filename CAIEQO), and "How to Pick a Knowledge Project From the 5 Choices" (filename KNOWPROJ). "How to Work With a Computer" has 43 slides and a 10-minute audio cassette. During the presentation, the students learn slide projector, cassette, and computer use. The Tandy Author I disk contains a program called "SAMPLE" that teaches how to take a test on the computer.

A second Author I test, which I wrote, asks them to look at the slides or listen to the tape and decide what problem is illustrated by each of the 20 examples. Disks appear upside down. Cigarettes and food menace disk surfaces. Contrast and brightness controls demand adjustment.

I use a four-student group and self-teaching methods. One student operates the slide projector, another the cassette, the third the computer, and the fourth is a participant/observer.

The KNOWPROJ module takes up the second hour of this two-hour lab. Eighty slides and a 14-minute tape explain the five Knowledge Project choices. The workbook shows them how to take a Quick Quiz. During this two-hour period, I spend a total of about 10 minutes with the four students.

There are two other two-hour CAI sessions in the lab: S/OEVL and OPINE.

The goal of Self/Other Evaluation (S/OEVL) is to let the neophite DJs see and hear problems they might encounter on the air. The lesson is broken into two parts: DJEVAL1, covering items 1-13, and DJEVAL2, covering items 14-29. There are slide and audio examples of each problem.

At the end of the presentation, the four-student team does an actual DJ evaluation, which is passed on to the student management and to the DJ. For their third and fourth times on the air they evaluate themselves and others in their group using the DJEVAL form.

Every student in Radio Production writes and records an editorial opinion on a campus or community problem. The Editorial (OPINE) goes on the air during their fourth 25-minute air shift.

QUICK QUIZ OPTION LIST

1 CREATE A FILE OF QUESTIONS

2 RUN A TEST ALREADY ON FILE

3 REVIEW AND EDIT A TEST ALREADY ON FILE

4 REVIEW TEST RESULTS

5 ERASE TEST RESULTS FROM DISKETTE

6 PRINT OUT TEST ON LINE PRINTER

7 END PROGRAM

***Figure 1.** Quick Quiz main menu.*

RADIO SHACK TRS-80 AUTHOR I
Lesson Development Program

The next page will be page #01

Press ENTER to add a TEXT page

OR PRESS

Q to add a QUESTION page
G to add a GLOSSARY page
C to add a CONTROL page
X to EXIT add mode
Enter selection:

***Figure 2.** Four basic types of screens make up Author I*

Just before they seat themselves at the computer console for the OPINE CIA unit, they've been in a production studio to record a standard introduction and conclusion to their editorial. When they finish the computer tutorial, they should be able to mix the voice track (words of the introduction or conclusion) with the standard musical theme used to identify an editorial. Upon completion of the 20-question Author I quiz, I give them a brief demonstration and let them get to work.

All 160 students in Radio Production have six hours of lab time devoted to CAI, and 70 percent of them spend another 2-3 hours on their own working on one of the knowledge projects featuring the computer.

Why have I been so long in getting to the "meat" of the article—the computer programs. Because the computer, if it is to be used successfully, has to be integrated into a total learning system that's comfortable for the teacher to work with and rewarding for the student. The computer is not THE solution; it's only another tool to help in the educational process.

Tandy's Quick Quiz is an inexpensive program ($39.95) that facilitates writing, taking, and scoring multiple-choice tests. The computer has an advantage over in-class paper and pencil testing: it can provide immediate reinforcement of correct or incorrect answers and immediate grades.

That's exactly what the Audio and Recording Techniques knowledge projects needed. The Audio series is a commercial filmstrip package, complete with narration on six cassette, six 20-question quizzes, and six filmstrips. To adapt it to CAI, I simply typed the questions into the computer and mounted the filmstrips into slide mounts to save wear and tear on the medium.

Basic is the language of the Quick Quiz. To get into the program, type in BASIC, wait for that to load, and then type in RUN"QQ". That command takes you to a menu that asks you what you want to do (see figure 1).

After typing 1 and hitting ENTER, the screen asks: HOW MANY QUESTIONS? After you reply, the screen queries: HOW MANY CHOICES? Type the number you've chosen and hit ENTER. The heading on the next screen reads: QUIZ CREATOR-PLEASE TYPE QUESTION 1.

Once you've completed the question —a maximum of three Model III lines— and hit ENTER, the program automatically inserts a 1 in the far left column. Type in the choices—up to two lines. At the end of the second line, hitting ENTER puts a 2 at the far left of the screen.

When you finish the fourth choice, the screen asks: WHICH CHOICE IS CORRECT? After you patiently inform the machine, it proposes one final step: QUESTION 1 OK? If it is, type Y (for yes) and the program stores the question. Y also takes you automatically to a new screen and a new question. If N (no) is your answer, the screen clears and you can redo the *whole* question

Everybody's making money selling microcomputers. Somebody's going to make money servicing them.

Now NRI Trains You At Home To Make Money Servicing, Repairing, and Programming Personal and Small Business Computers

Seems like every time you turn around, somebody comes along with a new computer for home or business use. And what's made it all possible is the amazing microprocessor, the tiny little chip that's a computer in itself.

Using this new technology, the industry is offering compact, affordable computers that handle things like payrolls, billing, inventory, and other jobs for business of every size... perform household functions including budgeting, environmental systems control, indexing recipes. And thousands of hobbyists are already owners, experimenting and developing their own programs.

Growing Demand for Computer Technicians

This is only one of the growth factors influencing the increasing opportunities for qualified computer technicians. The U.S. Department of Labor projects over a 600% increase in job openings for the decade; most of them *new* jobs created by the expanding world of the computer.

Learn At Home to Service Any Computer

NRI can train you for this exciting, rewarding field. Train you at home to service not only microcomputers, but word processors and data terminals, too. Train you at your convenience, with clearly written "bite-size" lessons that you do evenings or weekends, without going to classes or quitting your present job.

Your training is built around the latest model of the world's most popular computer. It's the amazing TRS-80™ Model 4, now with disk drive and the capabilities and features to perform a host of personal and business functions. No other small computer has so much software available for it, no other is used and relied on by so many people. And it's yours to keep for personal and business use.

You get plenty of practical experience. Under NRI's carefully planned training, you even install a *disk drive* verifying its operation at each step. Using the NRI Discovery Lab® that also comes as part of your course, you build and study circuits ranging from the simplest to the most advanced. You analyze and troubleshoot using the professional 4-function LCD digital multimeter you keep to use later in your work. Then you use the lab and meter to actually access the interior of your computer... build special circuits and write programs to control them. You "see" your computer at work and demonstrate its power.

(TRS-80 is a trademark of the Radio Shack division of Tandy Corp.)

Same Training Available With Color Computer

NRI offers you the opportunity to train with the TRS-80 Color Computer as an alternative to the Model 4. The same technique for getting inside is enhanced by using the new NRI-developed Computer Access Card. Only NRI offers you a choice to fit your specific training needs.

Become the Complete Computer Person

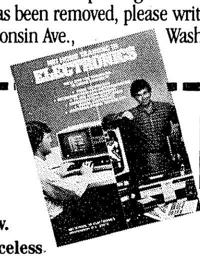
In addition to training in BASIC and advanced machine language, you gain hands-on experience in the operation and application of the latest computers for both business and personal jobs. You're trained to become the fully rounded, new breed of technician who can interface with the operational, programming, and service facets of all of today's computers. You're ready to take your place in the new electronic age.

Other Opportunities

NRI has been giving ambitious people new electronic skills since 1914. Today's offerings also include TV/Audio/Video Systems servicing with training on our exclusive Heath/Zenith computer-programmable 25" diagonal color TV... Industrial Electronics... Design Technology... and other state-of-the-art courses.

The Catalog is Free. The Training is Priceless.

Send the coupon for our 104-page catalog showing all courses with equipment and complete lesson plans. There's no obligation other than to yourself. See how NRI can help you take advantage of the exciting job and earning opportunities in the exploding field of microcomputers. If coupon has been removed, please write to NRI Schools, 3939 Wisconsin Ave., Washington, D.C. 20016.



NRI Schools
McGraw-Hill Continuing Education Center
3939 Wisconsin Avenue
Washington, D.C. 20016

We'll give you tomorrow.

All career courses approved under GI bill
☐ Check for details

The catalog is free. The training is priceless.
Please check for one free catalog only.

☐ Computer Electronics including Microcomputers
☐ Color TV, Audio, and Video System Servicing
☐ Electronics Design Technology
☐ Digital Electronics
☐ Communications Electronics • FCC Licenses • Mobile CB • Aircraft • Marine
☐ Industrial Electronics
☐ Basic Electronics
☐ Small Engine Servicing
☐ Appliance Servicing
☐ Automotive Servicing
☐ Auto Air Conditioning
☐ Air Conditioning, Heating, Refrigeration, & Solar Technology
☐ Building Construction

Name (Please Print) Age

Street

City/State/Zip

Accredited by the Accrediting Commission of the National Home Study Council

193-024

and choices.

When you finish typing all the questions, WHAT FILENAME DO YOU WANT TO USE FOR THE TEST ON THE DISKETTE? appears. An eight-character filename saves the file and returns you to the menu. You can end the program there (7), or go back and Review and Edit a test already on file (3), or take the test (2), or even print up copies (6).

The Quick Quiz program manual is only 25 pages long. The word processing and editing facilities seem limited. For instance, during the edit process you can change a question, but you have to type it all over again. The same limitation applies to the answers. But it's a very simple, very comfortable process, and it doesn't require memorizing a lot of codes. I've found that 15- to 20-question tests will fit comfortably on a disk.

For the Recording Techniques knowledge project, I broke a two-part filmstrip series into four parts (to fit within the 80 slides-per-tray limit): History, Recording Session, Disk Mastering, and Disk Duplicating. The filmstrip scripts were not as detailed as I wanted, so I wrote new scripts that went into the recording process in greater depth.

Technology has changed since the production of the original filmstrips. I wrote two more units—"Digital Audio Technology" and "The Long And Winding Road," a discussion of how hard it is to sell a record.

The CAI knowledge project choice is an audio tutorial tour of the Quick Quiz and Author I manuals. Using highlighters, I colored certain segments of the instruction manuals and referred to those lines by color on the tape. After cassette/manual instruction, the student creates a brief Quick Quiz and an Author I quiz on the computer.

Author I ($149.95) supports three basic types of questions: multiple choice (or true/false if those are the two choices); sliding cursor—where the student moves a blinking square over the correct answer and presses ENTER; and fill in the blanks. The program works on Models I, III, and 4. Four basic types of screens make up Author I (see figure 2).

The Text screen allows you to type up to 13 lines of 59 characters each. You can insert as many screens as you like (until memory overflows). A 48 KB Model III memory leaves about 28 KB after loading the Author module. My 20-question quizzes ranged from 14,000 to 19,000 characters. A double-spaced, 10-characters-per-inch page contains about 1,700 characters. Three units fit comfortably on a disk with score files. On the last Text screen, type in the question: multiple choice, true-false, sliding cursor, or fill in the blanks.

At that point you choose menu option Q, which brings up the Question screen. Actually, it should be called the Answer screen, because it's here that you tell the computer the correct answer.

At the same time, you can give positive reinforcement (good job, Joe), negative reinforcement (if you want to), and hints. If a student makes a particular kind of mistake (i.e., picks B instead of A), the computer flashes an immediate hint to guide the student to a better choice.

The Glossary screen allows the teacher to define terms. For example, you might type in "HINT-look up CUE if you're having problems. Press ?" in the hints

section for FCC Form 753. After the student types in the word CUE, the computer program displays the definition of the word. Glossary can be used at any time in the lesson.

Reference material can be added at any time, too. The TRS-80 Model III screen is 64 characters wide. Fifty-nine are reserved for Text, one for the vertical bar that shows the edge of the text segment, and four spaces for control codes. The codes reference previous portions of the text simply by giving a hint such as "Look at this" (screen changes to sections indicated by LAB1-LAB2), and then the screen comes back to the Text screen and "Now try again" appears.

CONTROL (C) does what it sounds like it might do—it keeps things under control by routing students for branching and specially coded activities.

With a CHAIN, students can be moved to material that's not included in the lesson but is on the disk or in another drive. The CHAIN could review a segment of a previous lesson or add Basic programs—(e.g., math, keyboard functions, animation, graphics) into the text at that point.

Record keeping is fairly complete in Author I. STUDENT is the module's name. The menu allows you to ASMDX. Each student can be identified by a first name, last name, and an optional password.

The score file shows percentages and times for the total lesson. Optional percentages and times for up to 10 selected subsections can be reviewed or printed out.

The STUDENT module also allows the instructor to LOCKOUT a particular student so he or she can't take the test again. You can also choose to OVERLAY, not show the previous score but only the new one, or you can instruct the computer to AVERAGE scores.

The GRAPHICS mode helps create simple drawings or special symbols to add to the text screens.

Let me give you a little advice: Start simple. Maybe Quick Quiz will do what you want. It's priced right, it's comfortable, and it's familiar. Then as you learn more or become more ambitious, you can expand into Author I.

If you're really ambitious you might try Pilot (usable on Model I, III, or 4), which is a full-fledged computer language. Pilot is an acronym for Programmed Inquire, Learning, Or Testing. The program manual is a fat volume of 350 pages, and includes instructions and examples using 11 direct commands (e.g., SAVE, LIST, BOOT), 18 program commands (e.g., ACCEPT, MATCH, WAIT), 16 edit commands (e.g., SEARCH, KILL, INSERT, HACK), a set of graphics commands that permit drawing, plus advanced graphics and programming commands. It's quite a complicated program.

Frankly, as a teacher who has 160 students to teach, I don't have time to learn the language. I like Author I and Quick Quiz. After all, why learn French if the waiter will explain the menu in plain, if slightly accented, English?

I'm still in the debugging stages. I encourage my students (with extra points towards a grade) to find and report errors or difficulties. After one semester of use, I'm checking slides, tapes, and computer screens to see what little touches I can make to upgrade the teaching units.

In another year, I'll run a fine-toothed comb through everything and revise the whole system. For me, the question asked in the first paragraph has been answered positively. The horizon beckons. How about teaching script writing using word processing? Implementing a graphics tablet to simulate equipment operation? A touch screen? A light pen? Who knows. But at least, as the Chinese proverb has it, I've taken the first step on the long journey.

Arthur C. Matthews is general manager of WVSS, the University of Wisconsin, Stout public radio station. In 1982 he received a grant to study integrating computers into his learning system.

OPEN UP AN ACCOUNTING DEPARTMENT FOR $395.00

GENERAL LEDGER integrated postings from A/R, A/P and Payroll. **Prints 13 detailed reports** • Company or departmental Income Statements • **Comparative financial statements** with current, YTD, budget, and last year (month and YTD) • Presents everything you, your bookkeeper, and your accountant need to know • G/L reconciles all accounts and maintains extensive, detailed audit trails • Trial Balance includes all transactions • **Flexible Chart of Accounts** • True double-entry bookkeeping • Master File capacity: 400 accounts • Monthly Transaction capacity: 1,000 with 200K diskette; 3,500 with 500K diskette; 7,000 per Megabyte with a hard disk.

ACCOUNTS RECEIVABLE provides **instant, on-line customer account information (both current and aged),** complete invoicing (open-item or balance forward) and statement capabilities on optional preprinted forms give your company a professional image • Quickly identify overdue accounts, speed collections, help control cash flow • Detailed and summary customer activity and aging reports • Produces 8 reports • Automatic periodic customer/client billing option • Itemized monthly transactions • Master File capacity: 400 Customers • Monthly Transactions capacity: 800 with 200K diskette; 3,500 with 500K diskette; 7,000 per Megabyte with a hard disk.

ACCOUNTS PAYABLE maintains complete vendor/voucher history and includes **check-writing** capabilities • **Current and aged payable reports** • Cash flow/cash requirements report • Prints checks with comprehensive check stubs • **Produces 11 reports and documents** • Automatic pay selection program allows payment by due date or by discount date • **Manual and automatic checkwriting** • Check register • Master File capacity: 400 Vendors • Monthly Transactions capacity: 800 with 200K diskette; 3,500 with 500K diskette; 7,000 per Megabyte with a hard disk.

PAYROLL—Be the office hero each week when the checks come out on time! • **Calculates payroll for every type of employee** (hourly, salaried, and commissioned) and **prints payroll checks** (with popular, comprehensive check stubs) with an absolute minimum of input • Maintains monthly, quarterly, and yearly totals for reporting in **multiple states** • **User-maintainable Federal, State, and local tax tables** • W-2 printing • 941 Reporting • Produces 10 reports • Master File capacity: 400 employees.

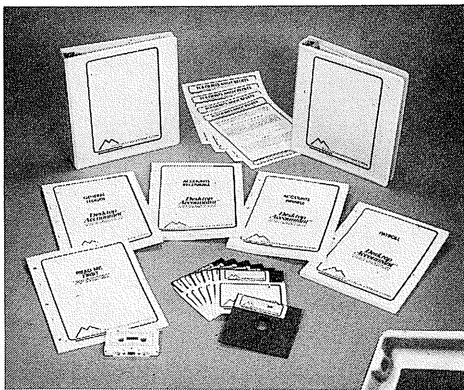
• California residents add 6½% Sales Tax • Payment by VISA/MasterCard/COD/MO/Cashier's Check • All Brand Names are manufacturers' registered trademarks • No sales to Dealers • Foreign orders please call or write before ordering • © 1983 Rocky Mountain Software Systems.

MasterCard VISA

Why staff up? With the Desktop Accountant™, all the accounting help your office needs can be at your fingertips!

Open up a wide new range of possibilities for your microcomputer! No matter what type of business you're in, Desktop Accountant will let you manage the financial end of it more professionally than ever before.

A Complete System with Support. Desktop Accountant includes General Ledger, Accounts Receivable, Accounts Payable and Payroll programs, along with comprehensive user manuals and training aids. We've even prepared an audio cassette tape to make learning the system fast and fun. And our telephone "hotline" means personalized support whenever you need it.



Desktop Accountant's fully integrated accounting system is a complete package of software, training aids, manuals and user's newsletter.

Produces 42 Reports. "Keeping the books" has never been so easy! Desktop Accountant prepares every bookkeeping and accounting report your growing business requires: from invoicing to statements to aged A/R listings; from cash distribution to A/P checks to vendor activity reports; from **complete** payroll checks and stubs to W-2 forms; from the chart of accounts to balance sheet and income statement, as well as many others so vital to efficient management.

Desktop Accountant is available for nearly every portable, personal and desktop computer. The system requires either CP/M® or MS-DOS™ (PC-DOS), Microsoft BASIC™, 64K RAM, two disk drives or hard disk, and a 132-column printer (or an 8½" x 11" printer with compressed print mode).

You won't find better quality software at such a low price—a price we can offer now because development costs were recovered years ago. Just **$395.00 for most CP/M® formats ($495.00 for IBM® and some CP/M formats) complete.** *Call for available formats.*

Order Desktop Accountant today! You'll soon see why we call it **INTELLIGENT SOFTWARE FOR INTELLIGENT PEOPLE.**

ROCKY MOUNTAIN SOFTWARE SYSTEMS
Business Microcomputers and Software

To order Desktop Accountant and for more comprehensive literature, call toll-free

☎ 1-800-832-2244
(In California call 1-800-732-2311)

or send orders to
DESKTOP ACCOUNTANT
1280-C Newell Avenue, Suite 1216
Walnut Creek, CA 94596

Assembly Language Instructions

by William Barden, Jr.

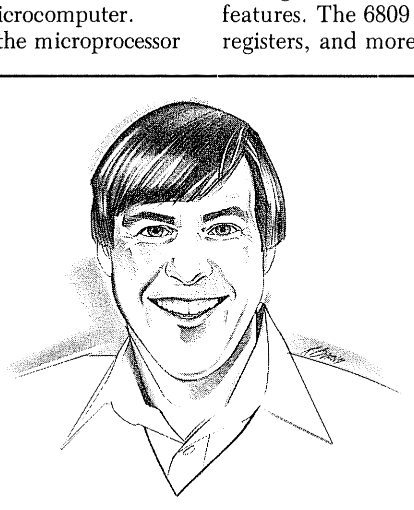
This month we're going to take a look at the instruction sets of the Z-80 microprocessor used in the Radio Shack Models I, III, and 4, and the 6809 microprocessor used in the Color Computer. Believe it or not, the instruction sets of all microprocessors are similar. To a large extent, if you learn how to program in Assembly language on one microprocessor, you'll be able to easily program in Assembly language on another microprocessor.

As we've mentioned previously, all Assembly language on Radio Shack computers is determined by the built-in instruction sets of the microprocessor contained in the microcomputer. There are "architectural" differences in how the microprocessor is used within the microcomputer, of course, and these affect things like the memory "map," the amount of RAM (random access memory) and ROM (read only memory), and input/output devices (line printers and the like). However, it's Zilog (Z-80) and Motorola (6809) that really determine which instructions to put into the microprocessor, not Radio Shack.

Instruction sets for computers have been geared towards arithmetic—addition, subtraction, shifting, and comparisons (subtraction without changing the results). Computers started off as extremely fast adding machines, and that orientation has remained through the Z-80 and 6809.

The arithmetic orientation of computers has carried over into most microprocessors. Just about all popular microprocessors have arithmetic instructions as a base; you can still add, subtract, shift, and compare numbers, and there are plenty of instructions to perform these operations.

Another strong influence on the instruction set is the tendency to make the microprocessor "downwards compatible," to resemble prior microprocessors produced by the semiconductor manufacturer. As a result, the Z-80 instruction set includes instructions used on the Intel 8080 (Zilog engineers came from Intel). And the Intel 8080 includes instructions used on the Intel 8008 and instructions similar to those used on the 4004 in 1976! The 6809 instruction set includes instructions used on the 6800, an older Motorola microprocessor. Downwards compatibility makes the microprocessor easier to sell to engineers who are using the manufacturer's existing products.



Assembly language programming is easy after you master it on any microprocessor.

The trend has been to make the new microprocessor similar to older products, but to add enough "bells and whistles" to make the new microprocessor more desirable. It sounds like selling soap or new cars, doesn't it?

The Z-80 added powerful "block move" and "block compare" instructions to the basic 8080 instruction set, along with adding additional registers, "interrupt" capability, and other features. The 6809 added a Multiply instruction, additional registers, and more powerful "addressing" to the basic 6800 instruction set.

At first glance, it appears as if the instruction sets of the Z-80 and 6809 are hundreds of separate instructions. Many of the instructions, however, are similar, and on both microprocessors, the instruction sets can be easily organized into only a few groups.

In the following discussion, we'll describe some of the most commonly used instructions in the Z-80 and 6809, and leave the more esoteric instructions for practical examples in future columns.

Adds in the Z-80 and 6809 add either two 8-bit numbers or two 16-bit numbers. One of the numbers is always in an "accumulator," while the other operand is always from another register or from memory.

The flags (Z-80) or condition codes (6809) are set according to the result of the add, as we described in the last column. (The flags can be tested in "conditional jump" instructions that let you alter the path of the program based on the results of the add.)

Z-80: The result goes into the A accumulator (8-bit) or the HL, IX, or IY register (16-bit). If the result is too large to be held in 8 or 16 bits, the result still goes into the A or HL register, but the P/V flag (parity/overflow flag) is set to indicate overflow. The following are some typical Z-80 adds:

```
ADD    A, B     ;add contents of B to A
ADD    A, 34    ;add 34 to A
ADD    HL,BC    ;add BC register pair to HL
ADD    IX,BC    ;add BC register pair to IX
ADD    IY,BC    ;add BC register pair to IY
```

6809: The result goes into the A or B accumulator (8-bit) or into the D register (A and B accumulator combined as one 16-bit register). The V (overflow) condition code is set if the result is too large to be held in 8 or 16 bits, but the result is still loaded. The "ABX" instruction is unique—it adds the contents of the 8-bit B register to the contents of the 16-bit X index register. The following are some typical 6809 adds:

```
ADDA   #34      add 34 to A
ADDB   $100     add contents of RAM $100 to B
ADDD   #1240    add 1240 to D register
ABX             add B register to X
```

Subtracts are just like adds in that they set the flags or condition codes and operate on either two 8-bit or two 16-bit numbers with the result being put into an accumulator. The operand in the accumulator is always the "minuend," the number from which the other operand is subtracted.

Z-80: 8-bit subtracts operate on the A register; 16-bit subtracts operate only on the HL register, and not on the IX or IY registers. Here are a few typical Z-80 subtracts:

```
SUB    34       ;subtract 34 from A
SUB    B        ;subtract B register from A
SBC    1234     ;subtract 1234 from HL
```

6809: 8-bit subtracts operate on either the A or B registers, while 16-bit subtracts operate on the D register. Here are a few typical 6809 subtracts:

```
SUBA   #34      subtract 34 from A
SUBB   $100     subtract contents of RAM $100
SUBD   1240     subtract 1240 from D
```

The 6809 also has a special instruction that allows you to add or subtract a constant from the X, Y, U, or S registers, with the result going into the same or a different 16-bit register. It's called "LEA" (Load Effective Address):

```
LEAX   -1,X     subtract 1 from X, result to X
LEAX   100,Y    add 100 to Y, result to X
```

In addition to simple adds or subtracts, both the Z-80 and 6809 have adds or subtracts with "carry." The term carry means either a carry or "borrow" in microprocessors. These instructions allow you to string together several adds or subtracts in a chain of multiple-precision operations, letting you work with numbers greater than 8 or 16 bits. You could use four separate adds, the first a "normal" add and the next three adds with carry, to work with 32-bit numbers, for example.

Z-80 instructions using the Carry flag:

ADC A, SBC A, SBC HL

6809 instructions using the Carry condition code:

ADC, SBC

Both the Z-80 and 6809 have instructions that allow you to add or subtract one from the contents of an 8- or 16-bit register. These operations are so common that they merit an instruction by themselves. The resulting instruction usually takes up less memory and is faster than adding or subtracting one by using a variable from memory or an "immediate" value in the instruction itself. These instructions are called increments and decrements.

Z-80 increments and decrements:

```
INC   C     ;increment C register by 1
DEC   H     ;decrement H register by 1
INC   IX    ;increment IX register by 1
```

6809 increments and decrements:

```
INCA         increment A register by 1
DECB         decrement B register by 1
INC   $100   increment RAM location $100
```

The 6809 has a built-in multiply instruction called MUL, which multiplies the contents of the A register by the contents of the B register and puts the 16-bit result in the D register (A and B combined). The Z-80 has no built-in multiply instruction.

Both the Z-80 and 6809 have compare instructions. Compare instructions are identical to subtract instructions except that the result is not put into any register. This allows you to compare the contents of the register. The flags (Z-80) or condition codes (6809) are set on the result of the comparison and can be used in conditional jump instructions.

Z-80 compare:

```
CP    34    ;compare A with 34 (A-34)
```

6809 compares:

```
CMPA  #34     compare A with 34 (A-34)
CMPB  $100    compare B with contents of RAM
CMPD  #1000   compare D with 1000 (D-1000)
```

Shift instructions in the Z-80 and 6809 operate on 8-bit registers or memory locations. Shifts move the 8 bits of data 1 bit over into the next bit position. The flags or condition codes are set on the result of the shift. The most important flag affected is the Carry flag, which is set if a bit is shifted out of the register or memory location. This allows you to test a bit at a time, rather than the entire 8 bits.

There are three types of shifts in the Z-80 and 6809: rotates, logical shifts, and arithmetic shifts (see figure 1).

Rotates shift the 8 bits, but the data is shifted back into the register or memory location, sometimes via the Carry flag, in effect making the rotate a "9-bit" rotate.

Logical shifts move the 8 bits out of the register or memory location with the bit shifted out being lost (actually the Carry flag holds the bit until the next operation affecting the carry). A zero fills the vacated bit position.

Arithmetic shifts "sign extend" the value. The sign bit in the most significant bit, bit position 7, is preserved.

Z-80 shifts:

```
RLCA         ;rotate A left
RLC          ;rotate A left through carry
RRA          ;rotate A right through carry
RLC   (HL)   ;rotate RAM location left
SRL   B      ;shift B right logical
SRA   C      ;shift C right arithmetic
```

6809 shifts:

```
RORA          rotate A right
ROLB          rotate B left
ROL   $100    rotate RAM location $100 left
LSLA          logical shift A left
ASR   RESLT   arithmetic shift location RESLT
```

Jumps and branches are used all the time in both the Z-80 and the 6809. An "unconditional" jump is identical to a Basic GOTO. It loads the PC register with the jump address, causing

a new segment of code to be executed. A "conditional" jump tests the conditions of a previous add, subtract, compare, shift, or other instruction. The conditions are represented by the flags or condition codes. If the condition is met, the jump is made and a new set of instructions is executed. If the condition is not met, the jump "falls through" and the next instruction in sequence is executed.

Both the Z-80 and 6809 have "relative" jumps in which the instruction is shorter because it uses "relative addressing," referencing the jump address to the location of the jump instruction itself. The rationale for this is that conditional jumps are used so often that reducing the number of bytes in the instruction from 3 to 2 will save a great deal of program memory space and time.

Z-80 jumps:

```
JP    LOCN   ;unconditional jump to "LOCN"
JP    C,LOCN ;jump if Carry flag set
JP    Z,LOCN ;jump if Zero result
JR    Z,LOCN ;relative jump if Zero result
```

6809 jumps and branches:

```
JMP   LOCN   unconditional jump to "LOCN"
BRA   LOCN   unconditional relative "branch"
BNE   LOCN   branch if "not equal" (Z=0)
LBNE  LOCN   "long branch" if not equal
```

A subroutine in Assembly language is identical in concept to a Basic subroutine—it can be anything from a few instructions to thousands of instructions, grouped in one spot to save memory space. A CALL (Z-80) or JSR (6809) "calls" the subroutine in a form of unconditional jump. In this case, however, the return address of the instruction after the CALL or JSR is put into the "stack." At the end of a subroutine, an RET (Z-80) or RTS (6809) retrieves the return address and causes a return to the instruction following the "call." The whole operation is similar to a Basic GOSUB. . .RETURN sequence.

The stack is used not only for storing return addresses for subroutines calls, but also for storing temporary data (see last month's column). Since there are only so many CPU registers, the stack is often used to supplement the CPU registers to store temporary results. "Push" instructions store data on the stack and adjust the stack pointer to point to the next location in the stack. "Pull" or "pop" instructions retrieve the data from the stack.

Z-80 stack instructions (PUSH pushes the contents of a register pair—AF, BC, DE, HL, IX, or IY—onto the stack. POP pulls the data from the stack):

```
PUSH  AF     ;save the A register and flags
.
.
.
POP   AF     ;restore A and flags
```

6809 stack instructions (PSH pushes any combination of registers onto the stack. PULS pulls the data from the stack.)

```
PSHS  A,B,X  save the A, B, and X registers
.
.
.
PULS  A,B,X  retrieve the registers
```

A "load" instruction loads a CPU register with the contents of another register or memory location. A "store" puts the contents of a register into a RAM memory location. Both the Z-80 and 6809 have 8-bit and 16-bit stores.

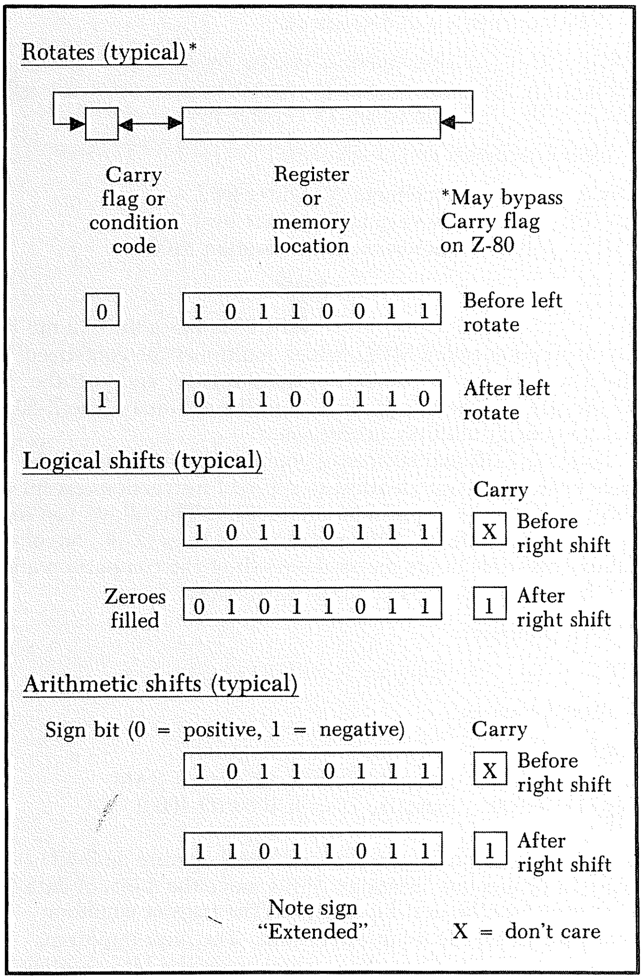


Figure 1. Types of shifts.

Z-80 loads and stores. (The mnemonic for Z-80 loads is LD. Confusing as it is, the mnemonic for a store is also LD. You can tell which direction the transfer is made by which operand comes first):

```
LD    (8000H),A ;store A to RAM location 8000H
LD    B,C       ;load B register with C
LD    (HL),B    ;store B using HL as a pointer
LD    HL,1000   ;load HL with value 1000
```

6809 loads and stores:

```
LDA   #34      load A register with 34
LDB   $100     load B register with RAM $100
LDX   #1234    load X register with 1234
STA   LOCN     store A into RAM "LOCN"
STY   CONST    store Y into RAM "CONST" (2 bytes)
```

The above instructions are some of the most commonly used Z-80 and 6809 instructions. We'll cover the remaining instructions as we encounter them in practical programs. The instructions just discussed probably represent 90 percent of the instructions used in a typical program.

Next month we'll talk about addressing modes, and then we'll be doing more in the way of pratical examples for both Z-80 and 6809 programs.

William Barden, Jr. is a full-time computer consultant and writer. He has worked in the computing industry for 20 years on a variety of systems, ranging from mainframes to microprocessors.

THE WAIT IS OVER

CP/M®

EXCLUSIVELY FOR THE

TRS-80® Model 4

Now, for the first time, unleash the powerful features resident in your Model 4 computer. Open up the vast store of CP/M software such as WordStar®, dBASE II and Multiplan™, along with thousands of others.

- Includes INTERCHANGE ™, a utility that allows reading, writing and copying 20 different manufacturers' disk formats such as IBM, KAYPRO, OSBORNE, XEROX, etc.
- Includes MEMLINK™, a unique feature that uses the optional 64K RAM memory as a fast disk drive.
- Complete with all these CP/M utilities; ASM, DDT, DUMP, ED, LOAD, PIP, STAT and SYSGEN.
- Operates at the 4Mhz clock in the standard Model 4 mode.
- NO HARDWARE MODIFICATIONS. Just insert the disk and boot.
- NO COPY PROTECTION. Backups may be made for your own use and protection.
- The CONFIGURATION program supports a full range of 5-1/4" disk drives: 35, 40, 77 and 80 tracks, single and dual sided in any combination as well as the standard Model 4 drives.
- Includes MODEM 7, a powerful public domain communications program for file transfer and remote data base access such as CompuServ and the Source.
- Supports 80 x 24 video, reverse video, direct cursor addressing and more.
- Utilizes the Model 4 function keys and allows user defined keys.
- Auto Execute command for turnkey applications.
- FORMAT utility permits up to 52 disk formats to be constructed, all menu driven.
- Fast backup routine with verify for mirror image copies.
- All support programs are menu driven for ease of use.
- Ready to run in the standard 64K Model 4. The additional, extra cost, 64K RAM upgrade not required.
- Complete with over 250 pages of comprehensive user documentation.

AVAILABLE NOW FOR IMMEDIATE SHIPMENT.......................................$ 199.95

The full line of MicroPro software is now available formatted for the Model 4 using our CP/M. Each disk is already configured and ready to run. Just install the printer of your choice and go.

WordStar® Fast memory mapped version	$250
MailMerge® Multi-purpose file merging program	125
SpellStar® 20,000 word proofreader on a disk	125
StarIndex™ Creates index and table of contents	95
WordStar Professional. All the above for only	450
InfoStar™ Advanced DBMS	$250
ReportStar™ Report generator & file manipulator	175
DataStar™ Data entry and retrieval package	150
SuperSort® Fast and flexible sorting is yours	125
CalcStar™ Advanced electronic spreadsheet	95

ORDER INFORMATION

Call now and your order will be shipped at once from our Dallas warehouse. We accept American Express, MasterCard, Visa, and most any other form of payment known to man. Credit cards are not charged until your order is shipped. Add $4 UPS surface shipping and handling on orders within the 48 States. No State Sales Tax on software or shipments delivered outside of Texas. No refunds. Defective items are replaced upon return, postpaid.

ORDER NOW . . . TOLL FREE

800-527-0347
800-442-1310

The Toll Free lines are for orders only.
Specifications subject to change without notice.

128K MEMORY UPGRADE

Our upgrade includes 64K of 150nsec RAM, genuine PAL® chip and instructions for installation. This kit will upgrade your 64K Model 4 to 128K and allow the use of our MEMLINK and TRSDOS 6.x MEMDISK. Comes with a full 1 year guarantee.

A BARGAIN AT ONLY $99.95

AD #883

MONTEZUMA MICRO

214-339-5104
Redbird Airport, Hangar #18
P.O. Box 32027
Dallas, Texas 75232

"WE KEEP YOU RUNNING"

VISA MasterCard AMERICAN EXPRESS

A Data Base Management System for Non-programmers

The InfoStar package—including DataStar and ReportStar—handles the capturing of data in addition to reporting on it without the user writing a lot of programs or purchasing a lot of separate software.

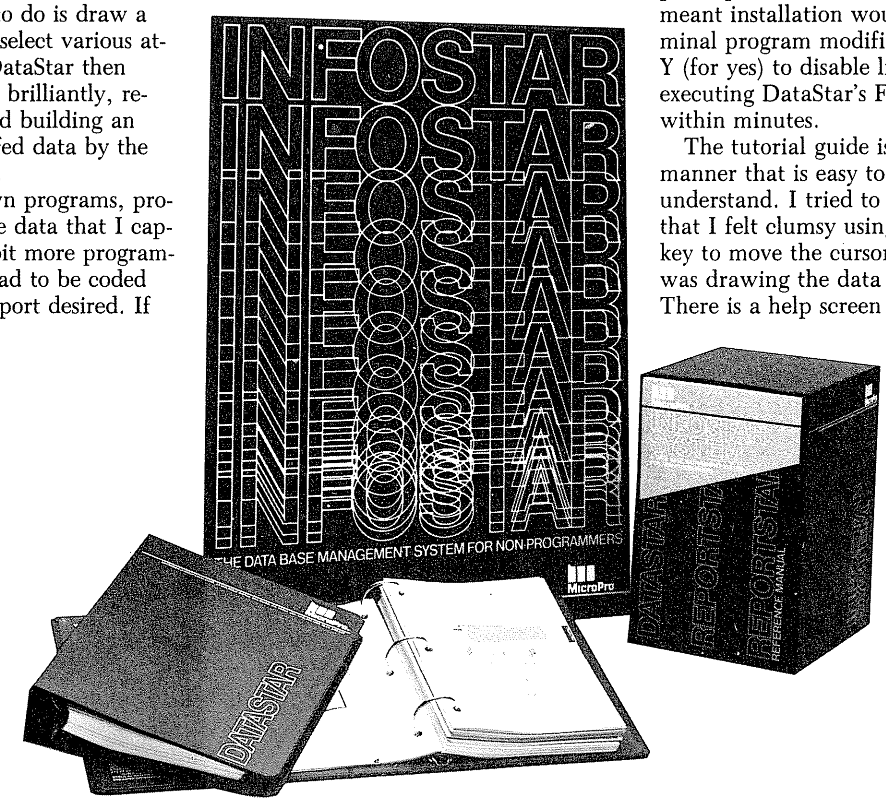
Reviewed by Patricia Steele

I have spent many tedious hours designing data entry screens and coding boring, but necessary, input verification routines. Even for the simplest report-producing programs, coding is needed to eliminate redundant key strokes at data entry time. If data files contain a large number of records, index routines are necessary for fast retrieval of the records. The end result of all this coding is a program that just places data in a file. The next chore involves coding a program that will delete unwanted records, update data fields in a record, add records to the file, and reorganize the file.

DataStar claims that it can do all these things; all the user has to do is draw a form on the screen and select various attributes for the fields. DataStar then takes over and performs brilliantly, retrieving, calculating, and building an index data base as it is fed data by the operator (see figure 1a).

When I coded my own programs, producing a report with the data that I captured required quite a bit more programming. A print routine had to be coded for every variation of report desired. If totals and breaks were wanted, that would take a few more hours of coding. The retrieval of data in Basic required a FIELD statement, conversion of the data into useful numeric data, an end of file routine, and a code to manipulate the data. If the file was created in alphabetic sequence and the report was needed in vendor number sequence, the coding to invoke a sort had to be done.

ReportStar boasts that it is more than just a report generator; it can produce anything from a simple list of information in a file to a complex, sophisticated report printed in any desired sequence (see figure 1b).



The descriptions of DataStar and ReportStar sounded very promising. I was ready to use the training guides and follow the instructions. I wanted to learn how to use these two "stars" so I'd never have to code a data base program again.

The first step, of course, was installing the software package. I did not use PIP to copy DataStar to my CP/M system diskette as instructed in the manual. With one floppy drive, I'd have been swapping diskettes four times for each program and file. I tried FastCopy, and it kept swapping to a minimum. I invoked DINSTALL and was happy to see TRS-80 II (Pickles & Trout CP/M) appear upon the third menu screen. That meant installation would require no terminal program modifications. I replied Y (for yes) to disable line feeds, and was executing DataStar's FormGen program within minutes.

The tutorial guide is written in a manner that is easy to follow and understand. I tried to ignore the fact that I felt clumsy using CTRL plus a key to move the cursor around when I was drawing the data entry screen. There is a help screen in view at all

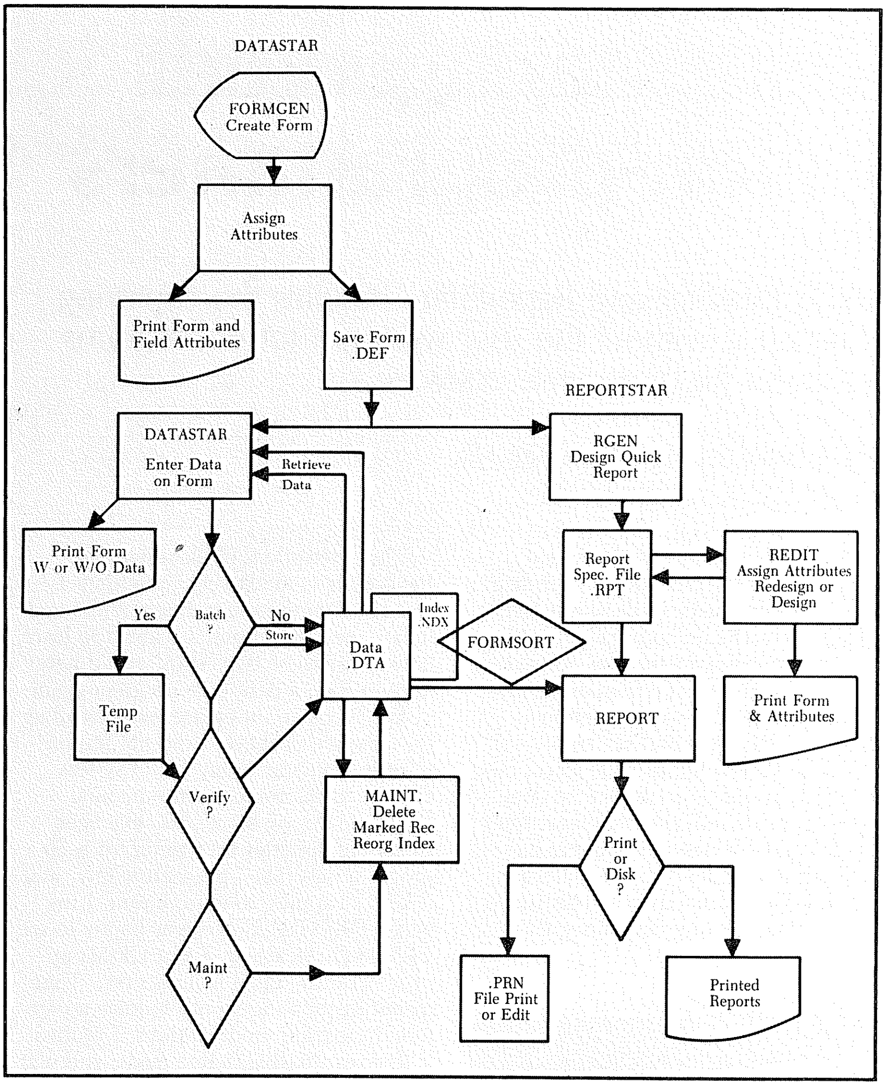


Figure 1a. *Figure 1b.*

times, however, so reference to the manual or the supplied command card is not necessary. After the CTRL keys are memorized, the help screen can be cleared and called back whenever needed. I still prefer the arrow keys, though. This minor inconvenience certainly wouldn't discourage my desire to purchase this package, but the arrow keys are there so why not make use of them? Maybe MicroPro can come up with a patch to allow it.

After drawing the suggested data entry screen, the guide will instruct that it be saved. An error message will appear, instructing that a key field was not selected and an opportunity will be given to select one. This is a good feature of the guide. It purposely instructs you to make mistakes as you are learning, informs you why the error happened, and tells you how to rectify it.

At least one field must be selected as the key field. This is accomplished by moving the cursor to the desired field and depressing the CTRL and K keys. When the form is passed to DataStar for data entry, DataStar will build an index file along with the data file. When a record is retrieved at report time or update time, the index is searched. The index contains a pointer to the record that is wanted, eliminating a sequential search of the file. This is referred to as the direct access method of fast retrieval.

DataStar goes one step further with its direct access method by allowing files created with FormGen to be used for "look up" tables. They can either be loaded into memory (RAM) and kept there at RUN time, or read when needed. After I graduated from the tutorial guide, I created an inventory screen (see figure 2) and chose to have four "look up" tables in RAM and one to be read as needed. I could hear the disk read as it brought the data to the screen, but because the file is indexed, it was fast. If all five files were read in as needed, five reads would be necessary and processing would be slowed down. The number of files kept in RAM depends upon the size of the files and the amount of RAM the computer has.

The package is shipped with sample forms and files that are used throughout the training guide. This lessens the time needed to learn how to retrieve data from files to fill the data screen with calculated and preset information.

Attribution time is where the most thought will have to be put into the retrieval of data. The most important consideration is to eliminate redundant key strokes during data entry. FormGen makes it simple to assign most of the attributes by selection. The cursor is placed over the field and a CTRL R is entered. It is menu driven from there on. FormGen allows a backward scroll, a forward scroll, and a skip out. Any errors will be noted and the cursor will be positioned for correction. I admit I spent most of my time re-reading and going over this section. It's the most important and powerful part of FormGen.

The attributes that can be assigned are as follows—they are all optional:

1. Field name
2. Field order
3. Key order
4. Copy attributes of field
5. Field derived
6. Right justify
7. Pad field characters
8. Floating characters
9. Verify field
10. Check digit
11. Range check
12. Edit mask
13. Record edit characters

The FormGen program generates a "form" to be filled in by a process called "data entry," and data is entered by invoking DataStar. FormGen allows you to make changes to the form after it is created. You may get a hard copy printout of the form and a listing of the attributes assigned to it (see figure 3). FormGen, which is the program that creates the data entry screen from a copy of a form drawn upon the screen, is a cleverly written, well-documented program. It takes the length of the line that is drawn upon the screen and gives it a field position and length within a record, and lets you assign any kind of attribute to it that you could possibly need via menu selections. If it doesn't need to be numeric or alphabetic only, if you don't want a floating dollar sign and decimal point, if you don't want to pull the information in from another

```
PRODUCT CODE:     BRAND CODE:      TYPE CODE:      BAR CODE: 000000000000
PRODUCT:          BRAND:
                  TYPE:
DIST CODE:        DISTRIBUTOR:

SIZE CODE:        SIZE:                 UNIT PRICE:
                                        COUNTY TAX:
----------------------------------------------------------------------

STOCK ON HAND:                          AMOUNT:     .
                                        COUNTY:     .
                                        TOTAL:      .

                  DATE:       /  /
```

Figure 2. Inventory screen.

```
                            Q=required
                            C=check dgt
                            J=right just       * * D E R I V E D *
                            W=write ed c       .     LIST     CALC *****VERIFICATION****
                            O=oper entry       .                    .
FIELD NUM/NAME              R=range chk  PAD/  . INDEX ITEM         .             FILE VERIFY
    LEN LIN COL KEY E=edit mask FLOAT . FIELD  NUM ORDER .  ORDER    FILE NAME
                            .       .          .                    .
001/PR CDE
    001 000 014         . Q      E .           .                    . F  001    A:PRODCDE
002/BR CDE
    002 000 029 001 . Q      E .               .                    . F  002    A:BRANDCD
003/TYPE CD
    002 000 044         . Q      E .           .                    . F  005    A:TYPECD
004/BAR CODE
    012 000 058 002 . Q J    E . P0            .                    . S  009
005/PRODUCT
    005 001 009         .        .             . 001   002          . F  003    A:PRODCDE
006/BRAND
    020 001 031         .     O  .             . 002   002          .
007/
    022 002 031         .     O  .             . 003   002          .
008/DISTCD
    002 003 011         . Q      E .           .                    . F  013    A:DISTCD
009/DISTRIBUTOR
    020 003 031         .        .             . 008   002          .
010/
    001 005 011         . Q      .             .                    . F  004    A:SIZECD
011/SIZE
    006 005 023         .        .             . 010   002          .
012/
    005 005 050         . Q J    E .           .                    .
013/
    004 006 051         . Q J    E .           .                    .
014/
    003 010 017         .   J    .             .                    .
015/
    007 010 047         .   J    E .           .            007 N .
016/
    005 011 049         .   J    E .           .            006 N .
017/
    007 012 047         .   J    E .           .            008 N .
018/DATE
    002 014 025         .    WORE .            .            012 N .
019/
    002 014 028         .    WORE .            .            011 N .
020/
    002 014 031         .    WORE .            .            010 N .
```

Figure 3. Field attribute definitions

file, if you don't want to calculate it from other fields, and it if really isn't necessary that any data be keyed in, then you don't even have to bother selecting attributes for it. You are never stuck with your first choice; the form is updatable just like the data.

The program DataStar is the data entry program that builds the data file either by direct entry to the file, or by batch processing. Batch processing may be done at the host computer or at terminals. Verification of data can be done either while entering data or before the data is added to the main data base.

The data base can be viewed via the scan option by index or by data entry order. The data base can also be viewed, updated, and records deleted by using the select-by-key option. This is accomplished by placing the cursor on the field that you want to select and keying in the information that you want to search for in the file. Hard copy can be

printed with or without the form as background. Data can be printed without the form for preprinted forms.

DataStar is shipped with a well-written tutorial guide. Three practice chapters are included for first-time data entry operators. Also included with DataStar is a file maintenance program that will physically remove the records that are marked for deletion, and reorganize the file and index so retrieval is kept at the maximum speed.

The only bug I found in DataStar is that if the file you are going to retrieve data from is misspelled, you will loop. To get out of the loop, a cold boot is necessary, and you will have to invoke DataStar or FormGen to correct the name of the file.

ReportStar's installation program, RINSTALL, has a few more options to select from. I used the suggested default options for a first-time user and chose the teletype-printer option since neither my line printer nor my daisy wheel printers appeared on the menu screen. RINSTALL can be run over and over. After several installations, I selected the teletype and the backspace option to eliminate the double line printing of bold characters that occurred with my daisy wheel printer. For the present, I forgot about my line printer. I don't even care to describe the results of that test. I'll show you how REDIT let me get around that problem later in the article.

ReportStar includes sample files that are available for use with the training guide. It is also necessary to pull over some files from DataStar to follow the guide.

ReportStar is dependent on DataStar (using the FormGen program) to create the form definition files. Each chapter will produce a report by chapter number. The first chapter creates a Quick Report with the program RGEN (see figure 1b). RGEN displays all the field names of a chosen data file. The field names can be selected in any order for the report. One of the attributes that can be selected when using FormGen is to name the fields. These are the names that will be used as headings for the reports. If field names are not chosen, the field numbers will appear. The report can now be saved, and a program named REPORT actually does the printing.

ReportStar lets you "toggle" instruction messages on the screen. You can ask for five different kinds of summaries with a Quick Report: Record Count, Total, Average, Largest, and Smallest. FormSort can be used to sort by key, and using the Quick Report, you can ask for

```
TOMSINV REPORT LISTING WITH FIELD AND FILE ATTRIBUTE DEFINITIONS

                    FIELD NUMBERS
.PO 0
P
P                                TOMSINV REPORT
P                                  _1/_2/_3
P
. BR CDE = _4
. BAR CODE =              5
P           PRODUCT  BRAND                        DISTRIBUTOR
            ____6  ________________________7  ____________________8

                    CONTROL CHARACTERS
.PO 0
P
P                                BTOMSINV REPORT
P                                  _1/_2/_3 B
P
.  BBR CDE = _4 B
.  BBAR CODE =             5 B
P S         PRODUCT  BRAND                        DISTRIBUTOR S
            ____6  ________________________7  ____________________8
```

Figure 4. Report listing with field and file attribute definitions.

subtotals by the key field ID, as well as by page and by report. The reports are printed with the date keyed in at the report time.

As new data is added to a file, you will want updated reports. To get them, you simply call up the Report program and specify the filename without the .RPT extension. If you only have one floppy drive as I do, INVCE.DEF and INVCE.DTA will have to be copied and renamed from the ReportStar distribution disk to DataStar in order to use FormGen when changing the keys, and then copied back again to use FormSort. Doing it this way once was enough. I created a CP/M system diskette, deleted everything I thought I would not need, and FastCopied both DataStar and ReportStar to the one diskette.

ReportStar's program called REDIT is used to add to and change Quick Reports and design new reports. There are three parts to REDIT: File Description, Field Description, and Report Layout. Descriptions will print across the top of the screen, and help screens are available at all times. ReportStar can either take data from a file and put it into a report, or take data from a report and put it into a file. "Include If" expressions are used to set up a variety of conditions for reporting. This is where ReportStar becomes more than just a report writer.

When I read about the Special Effects and Print Control commands in ReportStar's training guide, I found a way to get around my printing problems. The first column of the layout is called the print control column (see figure 4). Then P tells Report to print once per report page. The period (.) tells Report not to print the report line. The other control characters show on the body of the report listing. I removed the Print Bold (B before TOMSINV and after the 3) and Underline (the S before PRODUCT and after DISTRIBUTOR). That took care of my line printer. Too bad REDIT won't let you change the .DEF forms created by FormGen.

The title of the report can be changed with REDIT, and a list of the instructions in the report specification file can be printed. The layout form can be 255 lines long and 254 characters wide, but I've never been introduced to a printer that prints more than 132 characters, the first position of a 133 byte buffer being used for forms control. I don't understand the logic there. The screen is only big enough to show a portion of that area. By typing beyond the right-hand boundary of the screen, the "window" will move to the right. By adding lines to the report, the "window" will move down.

Drawing a report layout with REDIT is similar to drawing a form with FormGen. Report fields must be long enough for the data that go into them. When the cursor is moved into a field, the status line will show the length of

the field. A CTRL F is used to define the fields. FormGen and REDIT are similar, but different—read the training guides of DataStar and ReportStar carefully.

Reports can be created without datafiles with operator input and calculated fields only. Backup files are created automatically. Data fields can be added to a datafile, and datafiles can be merged for a report. Numbers can be rounded off, and you can choose from several levels of error reporting. ReportStar can be used to print labels in two or more columns. The features just go on and on.

DataStar is designed to be used in conjunction with other products by Micropro. The files created by DataStar are compatible with WordStar, MailMerge, SpellStar, SuperSort, and CalcStar. DataStar can be used for many applications including legal document systems, inventory lists, a personal datebook, appointments schedules, and receivable systems.

SuperSort can be used to sort any file created by DataStar. You can write a letter with WordStar, and then use MailMerge to select the names and addresses from the DataStar files. If you use CalcStar to create an inventory file, you can access that file using DataStar.

You don't have to be a programmer to use DataStar and ReportStar, and you can start saving data within a short amount of time after installing the software. But time and patience will be needed to thoroughly understand assigning attributes before you can start printing those glorious reports. Once you're comfortable with the program, though, I can't think of an application that these two software packages couldn't handle. They are truly "stars."

Patricia Steele has been involved in the computer industry for 15 years. She is a professional (IBM mainframe) senior systems programmer turned microcomputer software programmer.

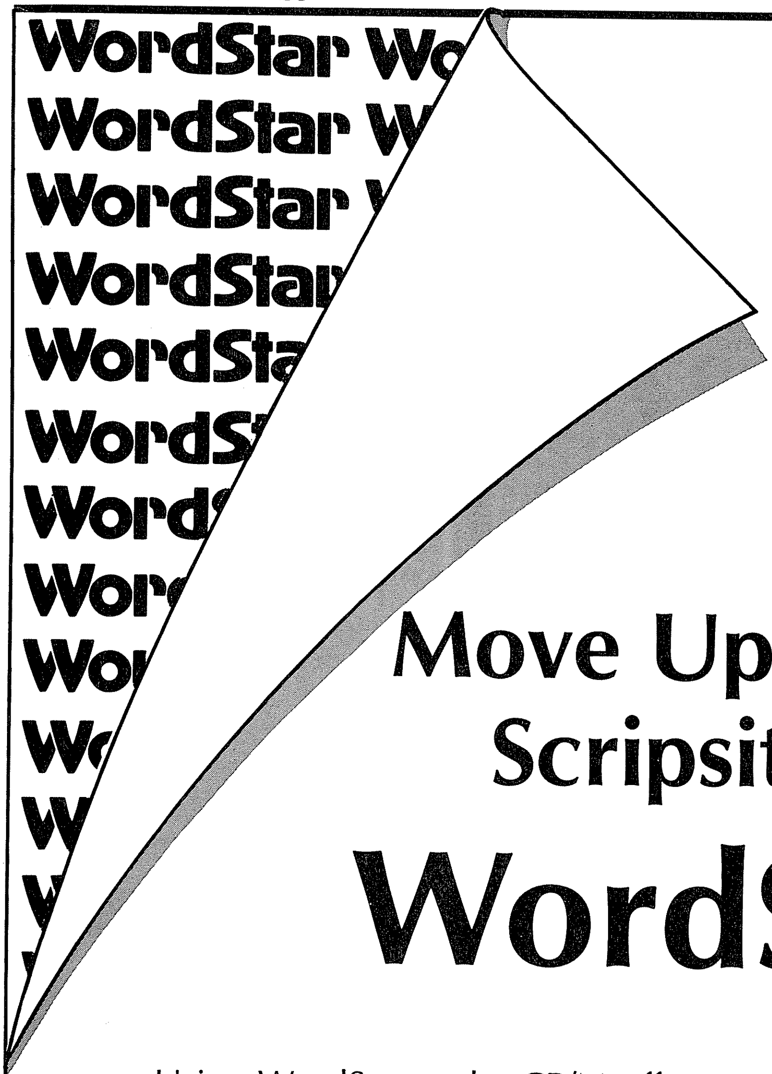
Product:
InfoStar (includes DataStar, ReportStar, and Manuals)

Manufacturer:
MicroPro International Corp
33 San Pablo Avenue
San Rafael, CA 94903
(415) 457-8990

System Requirements:
TRS-80 II/12/16 with CP/M operating system
Printer (optional)

Special Requirements: None

Price: $495



Move Up from Scripsit to WordStar

Using WordStar under CP/M offers owners of Radio Shack business computers an attractive alternative to Scripsit.

by Joseph Katz

This article was written on a Radio Shack TRS-80 Model II microcomputer using WordStar 3.0 (from MicroPro; San Rafel, CA) as the word processing software and Pickles & Trout CP/M as the operating system. I never use Radio Shack's TRDOS-based Scripsit anymore, which is ironic because a few years ago Scripsit 2.0 was what sold me on buying the Model II. It's even more ironic because WordStar was what originally sold me on Scripsit.

As a professional writer, editor, and consultant, I needed a serious word processing system that would handle everything from brief letters to book manuscripts. The system had to be reliable, flexible, and easy to use so I could concentrate on my writing instead of on the computer. As I studied what was available, I saw WordStar demonstrated on microcomputers from Apples to Zeniths. Its array of control codes—12 of them just to manipulate the cursor—seemed too demanding, too obtrusive. In contrast, Scripsit seemed just what I wanted. I bought it. Six months later I was using WordStar.

WordStar turned out to be less demanding than Scripsit. That conclusion will astonish Scripsit users who embraced it after seeing the kinds of WordStar demonstrations given to me. A salesperson would sit at a keyboard, ignore its arrow keys, and explain that to move the cursor just one character left, right, up, or down required two keystrokes each: ^S, ^D, ^E, or ^X. Since cursor movement is the single most frequent operation in word processing, that seemed to be madness. I fault MicroPro International, the makers of WordStar, and the dealers who demonstrated the program to me.

Everyone explains that WordStar should be *installed* for a particular system. Installation is a simple matter of running a program called Install and answering its questions properly. Then WordStar will work with the micro, the CP/M configuration, and the printer in the system. But the installed WordStar is a lumbering beast in a primitive environment. Since WordStar was designed to run on the widest possible range of CP/M machines, it assumes only the sparsest kind of keyboard. That's why control codes are used to move the cursor even when a keyboard has arrow and function keys, as Radio Shack's business micros have.

No one explains that after WordStar has been installed it should be *configured* to take advantage of a keyboard's features. Configuration is a simple matter of patching WordStar to replace its primitive cursor movement commands with functioning arrow keys. The accompanying sidebar shows how to do that for WordStar on the Models II, 12,

(Continued on page 65)

Phoenix

an advanced personal computer

Basic $599 kit (not shown) includes:

- Software compatibility with TRS-80 Model III and Model IV, plus CP/M
- 128k RAM card (64k normal plus 64k bank-selectable), less RAM
- 80 x 24 and 64 x 16 U/L case alphanumeric displays (software selectable)
- Z-80 CPU, with bootstrap ROM and hardware/software selectable 2 MHz and 5 MHz clock
- High resolution 512 x 256 graphics circuitry, with alphagraphics (less 16k high resolution RAM)
- Disk controller for any mix of up to four disk drives (5-¼"/8", single/double sided, single/double density, built-in/external)
- Parallel printer and light pen interfaces.
- Built in audio
- Provisions for readily available system ROM
- Tan polyurethane enameled metal enclosure, with power supply
- Standard typewriter keyboard, plus numeric keypad
- CPU board, with six expansion slots
- Parts kit, including ICs, sockets, fasteners and mounting hardware
- Assembly manual

Complete $1699 kit shown includes:

- High-resolution 12" green screen monitor
- Two SSDD slimline 5-¼" floppy drives and power supplies
- Hand rubbed, solid walnut end panels
- RS232 Interface board
- System ROM
- 128K system and user RAM
- 16k high resolution graphics RAM

Additional Options

- Single or dual built in slimline 5¼" SSDD or DSDD floppy drives and power supply
- Single or dual external slimline 8" SSDD or DSDD floppy drives, enclosure, power supply
- 128k bank selectable RAM board (for 256K total)
- 4164 RAMs
- Monitor
- Color Graphics (available 12/83)
- Light Pen (available 10/83)
- Hard disk host adapter (available 10/83)
- Factory assembled units (available soon)

PROGRESSIVE ELECTRONICS

537 East Main Street • Lancaster, Ohio 43130 • Tel: (614) 687-1019

TRS-80 is a Tandy trademark CP/M is a Digital Research trademark Dealer Inquiries Invited

CIRCLE NO. 43 ON INQUIRY CARD

How to Configure WordStar 3.0 for Arrow Keys on Radio Shack Business Computers

If WordStar has been previously installed and you know how to use DDT, the CP/M debug program, you may just be interested in the locations for patching to get the use of arrow keys. In that case skip to the listing of patch locations and codes. Just be sure to save the right number of pages for the patched WS.COM. Or if WordStar has been previously installed but you don't want to use DDT and just want to do the patching, read the next two paragraphs and then skip to the last paragraph. If, however, you've never used DDT or WordStar has never been installed, use all the following instructions.

The essential programs WSOVLY1.OVR, WSMSGS.OVR, INSTALL.COM, and—depending upon whether or not WordStar has been previously installed—either WSU.COM or WS.COM should be on a backup disk containing the CP/M system. Make sure the disk is not write protected. Put it in Drive A, the one Radio Shack calls Drive 0.

Execute the installation command by typing INSTALL <ENTER>. If WordStar has never been installed answer Y for yes when asked if you want a normal first time installation. At the Installation Options Menu choose option A. Then skip the rest of this paragraph and go on to the next. If WordStar has been installed previously answer N for no when asked if you want a normal first time installation. At the Installation Options Menu choose option B, C, or D (it doesn't matter which). Then at each succeeding menu simply press ENTER twice (for a total of eight times). Read the next paragraph, then skip the following four paragraphs and go to the very last paragraph.

Now pause for a moment and read the rest of this paragraph carefully. Patching any program has the potential for doing bad things to it. To minimize the risk, here are a few things to remember. If you want to abort the installation at any time before it is finished, hit the BREAK key (or key ^C). Nothing you did will be remembered by the program. If WordStar has been previously installed, there will be two questions after you make your selection at the Installation Options Menu. The first will ask, "Filename of WordStar to be INSTALLed?" Answer with the name you've used for WordStar (normally WS.COM). The second question will ask, "Filename for saving INSTALLed WordStar?" Answer with a nonsense name, not the same filename you are installing (GLURG.COM will do). That way, if anything is botched in patching, the original WordStar file is left intact and you can experiment with the nonsense file. If it works properly you can erase the original WordStar file and rename the nonsense file. Now we can proceed.

At Terminal Menu #1 choose option 3 to get Terminal Menu #3. When you have Terminal Menu #3, choose option [(the open square bracket) if you are installing WordStar to run on Pickles & Trout CP/M. For any other version of CP/M, select option % (the percent sign) and follow the instructions in the program. You'll be asked to confirm your selection just before WordStar shows you the next menu. If the selection is okay, answer Y and proceed. Otherwise answer N and re-enter the selection. (The installation program asks you to confirm each selection and, at the end, the result of all the selections. I won't mention the confirmation steps again. Just handle them the same way you did this one: Y if it's right, N if you want to change your mind.)

When you've confirmed your selection in Terminal Menu #3, you'll be shown the Printer Menu. The selection you make will depend on the printer you have. Both the Daisy Wheel II and the Epson MX-80 start with option C, 'Teletype-like' Printer that can BACKSPACE, but each requires further patching of WordStar to access all their features. The Epson requires still more patching to function properly. If you're not sure what to do, choose option A; just about every printer will work with that option.

When you've confirmed your selection in the "Printer Menu" you'll be shown the Communications Protocol Menu. The selection you make depends again on your printer. Many printers—among them the DWII and the Epson—need no communications protocol because they are fast enough to accept commands from WordStar in a stream. For these printers choose option N. If you don't know what to do, choose option N and see what happens the first time you start printing. The worst that can happen is that the printer will not be able to keep up. In that case all you have to do is experiment by reinstalling WordStar with different options until your printer behaves properly. If it's a reasonably fast, good quality printer, the chances are that it needs no communications protocol.

When you have confirmed your selection in the Communications Protocol Menu, you'll be shown the Driver Menu. The selection you make depends upon the printer and where the cable from it is connected to the computer. If the cable is connected to the parallel printer port, select option L, which works on Pickles & Trout CP/M and should work on other versions of CP/M (unless you changed the logical assignment using SETUP.COM, CONFIG.COM, or whatever your configuration program is called).

When you have confirmed your selection in the Driver Menu (or if you have retained the selections in a previously-installed WordStar command file by pressing ENTER eight times), you will be asked, "Are the modifications to WordStar now complete?" Answer N and you'll be given the opportunity you've been waiting for, the chance to use WordStar's patcher to change the values for controlling single-character cursor movement. Use the listing to get the four locations and the values to enter there, and follow the patching instructions in the installation program carefully. When asked for the location, just type the four characters under Location in the listing. When asked for the new code, just type the two characters next to them under Code in the listing. When you have finished the four changes enter a 0 for the next location to patch. Then you will be asked to confirm your options (answer Y). Finally your new WordStar will appear, with its arrow keys enabled. —JK

Arrow	Location	Code
Left	0495	1C
Right	049D	1D
Down	04A9	1F
Up	04AD	1E

Listing of patch locations and codes

Move to WordStar
(Continued from page 62)

and 16, no matter what CP/M you use on them.

Nor does anyone explain that WordStar can be fully customized to suit the user's preferences or whims. The dense *WordStar Installation Manual* shows a few patches that can be made to change some defaults (i.e., whether WordStar comes up in insert or overwrite mode), but what isn't clearly stated is that WordStar can be patched to change *all* its commands. The capability of being fully customized is WordStar's most magnificent feature. I've produced versions of WordStar for specific jobs and for each of my printers. Because Scripsit's commands are well suited to the Radio Shack keyboard, all of my WordStar versions are customized to emulate Scripsit's capabilities, as far as they go. WordStar can do more than Scripsit, though, so there are additional commands unique to WordStar. In most ways, a customized WordStar is easier to use than the menu-driven Scripsit. It's a sleek animal.

What made me realize I had to find another word processing package were experiences with Scripsit that led me to conclude it is simply too unreliable for serious word processing. From time to time Scripsit crashed. While I was using Scripsit 2.0, Radio Shack published several patches for it in *TRS-80 Microcomputer News.* I faithfully patched Scripsit each time, but some of the published patches turned out to cause even worse problems than they were supposed to solve. Eventually Scripsit 2.1 was released. By that time I had lost interest in experimenting with the security of my work.

I've had WordStar crash a few times, usually because I've done something stupid like prepare a long document on a disk that had too little space remaining. But WordStar crashes are nuisances, not disasters. The most that is lost is what is in RAM. Even that can be rescued by plugging in another disk and telling WordStar to put the file on it. WordStar files are in ASCII (American Standard Code for Information Interchange), and they are stored independently, outside WordStar. So WordStar can be on one disk, and the document on another.

WordStar provides even more safety. There's only one way to save a document in Scripsit: ESC Q, which closes the file and returns you to the main menu. That discourages the frequent saves needed for security. No one is going to pause periodically in a long document, close the file, get back to the main menu, open the document again, enter the password, wade past the "Open Document" menu, get to the page where they left off, move the cursor to the end, and start writing again. You don't have to do all this with WordStar. There are three ways to save a document: one saves and closes the file and brings you back to the main menu; another saves the file, closes it, and exits WordStar; and the third leaves you in the document while it is being saved, and two keystrokes bring you back to where you were when you made the save. It's so easy to make frequent saves that I do it whenever I pause for any reason.

There's a fourth WordStar command for saving that offers safety of a different kind. It allows you to quit the document without saving any of the changes made in it. Sometimes I return to something I've written and start "improving" it, only to realize that the original version was better. All I do then is tell WordStar to quit the document without saving it, and what I started with is left untouched—providing, of course, that I hadn't saved any of my "improvements" before quitting.

If I have made just one save before quitting, I can take advantage of another WordStar safety feature. Each time it saves a reopened document, WordStar writes the last version to a backup file. All I have to do is erase the document, rename the backup, and I have what I started with.

Not only is WordStar more reliable than Scripsit, it is also more flexible. It can do valuable things that Scripsit either can't do or won't do without taking a great deal of time and performing a great many contortions. For example, you can get an ASCII version of a document from Scripsit, but doing it requires conversion, which takes time and disk space. In the course of my work, I often communicate with distant computers. When I used Scripsit, the process of transferring files was a real pain. I'd have to convert my document into ASCII, using a second disk because Scripsit takes up so much disk space. Then I'd use Radio Shack's Terminal program, sometimes on a third disk, to send my document. If the other computer sent me a file in exchange, I'd have to reverse the process, converting from Scripsit to ASCII.

With WordStar there's no need for conversion. One of my program disks has both WordStar and a more sophisticated file transfer program than Terminal, so I've been able to receive short documents, exit the file transfer program while keeping the remote computer on line, edit the documents in WordStar, re-enter the file transfer program, and send the edited documents to the other computer. Such a process would not have been practical with Scripsit because it would take too much time.

Although neither word processor lives up to the boast of "what you see is what you get," Scripsit shows you a more limited picture than WordStar of how your document will look on paper. If you want to print the document with an even right-hand margin (justified), you'll see it with a ragged margin on the screen. WordStar, on the other hand, shows you the margin just as it will be printed. In addition, if you're doing something really complex, WordStar gives you the option of "printing" to a disk file instead of the printer, then viewing the file to see if you are getting just what you intended. If not, you can edit the document and print to disk again for checking. That can save an enormous amount of time and supplies. Scripsit doesn't do it.

WordStar is more flexible than Scripsit in preparing and editing a document, too. Limited boilerplating abilities built into WordStar remove some of the tedium from a day's work. Articles, for example, begin with a return address block. The last time I actually typed one, though, was a couple of years ago when I wrote my masterpiece, a WordStar file called HDR. All I do is use a WordStar feature that allows me to read one file into another and put HDR at the beginning of each article. It takes less than a second to do. Scripsit can "assemble" separate documents, but that takes longer.

I often use WordStar's ability to read and write parts of a document. Sometimes I come up with a few paragraphs that are great in themselves, but completely wrong for the article in which they appear. So I write them into a separate file, delete them from the current one, and use them elsewhere. You can't do that with Scripsit.

Simple mechanical things are also easier in WordStar than in Scripsit. Take, for example, the business of deciding the number of lines you want on a page. With Scripsit you make that decision when you create the document. Changing your mind requires spending some time waiting while Scripsit grinds away reshuffling things. Sometimes repagination caused Scripsit either to crash or to mess up the document by reformatting tabular material. I used to dread repagination. With WordStar it's a breeze—four or five keystrokes does it.

You see the page breaks instantly, and you can change them back and forth until you get just what you want. WordStar lets you force the ends of individual pages, just as Scripsit does, but with WordStar you have more flexibility. You can establish "conditional page breaks," to ensure that a page doesn't start with a "widow" (less than one or two full lines).

Are there *any* areas in which Scripsit does things WordStar can't, or does them better or more easily? Indeed there are, but only four seem at all worth considering.

WordStar is "document oriented"; Scripsit is "page oriented." That's why it's easier to change page breaks in WordStar: they have little significance to the program until it prints the document. But that means you can't search for a page by its number, as you can with Scripsit.

WordStar allows you to define only four special printing characters compared with the 26 you can define in Scripsit. That means Scripsit takes better advantage of the vast character set on printers like the DWII. It's not as great an advantage for Scripsit as it might seem, however, because WordStar has a way to get most accented characters by backspacing and overprinting the accent mark. If you're going to produce numerous things in French, German, and Spanish on a Radio Shack printer, and don't want to get a foreign language version of WordStar (yes, they are available), Scripsit has the edge. I use quotations from these languages occasionally, and WordStar does fine for me, but Scripsit does have the edge in printing special characters.

Scripsit also has the edge in providing 20 user-definable keys, each of which can store up to 26 keystrokes. Had I been writing this article in Scripsit, I could have put each word processing program's name in a different key, and with three keystrokes each, I could be entering the names as they came up. WordStar allows no user-definable keys. I miss them.

The fourth feature Scripsit has that WordStar doesn't is the ability to print multiple originals up to 255, in a single session. WordStar prints one original at a time. I've never had the need for two originals of anything at the same time, and I can't think of a situation in which an original and photocopies wouldn't do as well, so I never used this feature of Scripsit (except once to see if it worked).

What I do care about much more is the flexibility WordStar has gained by attracting programmers who have written utility programs for it. Because WordStar files are independent ASCII files, they are capable of being manipulated by external programs in ways that Scripsit files are not. And because WordStar is the bestseller of word processing programs for microcomputers, there are utilities for doing almost every conceivable thing with its files. As one kind of utility turns out to be useful and popular, it attracts competing programmers who produce even more elaborate versions. That result is that WordStar users can pick and choose to get just the tools they need—spelling checkers, mailing list programs, data base managers, and so on. It may seem odd to consider either a mailing list program or a data base manager as a word processing utility program, but in many applications that is all they are—accessories for the manipulation of text produced by the word processor. There are many more of them available for WordStar than for Scripsit.

WordStar provides a far broader horizon than Scripsit users ever see. For example, both WordStar and Scripsit let you print superscripts, but with Scripsit you can't use them for footnotes. You must put notes at the end of a document. Footnote, a utility for WordStar, allows you to use either endnotes or footnotes and even to change your mind over and over again.

Scripsit has an apparent advantage over WordStar in allowing you to number the lines in a document when you print it. That's useful in preparing a draft, because a committee can pick it to pieces by line number. WordStar doesn't give you that option. PIP, one of the standard utilities in CP/M, does allow this, but it numbers every single line including the blanks between double-spaced lines. There is, however, Enumerate, another WordStar add-on, which allows far more sophisticated line numbering than PIP. A script for a series of radio or TV commercials, for example, can begin each page with a new series of numbers, and the numbers can be placed just where you want them.

Once a document is completed, other utilities for WordStar provide the ability to give it finishing touches. Doing a table of contents and index in Scripsit, for example, is a manual operation in which you must refer to notes on what is where. But WordStar's Index utility computerizes the operations when all you need is a simple table of contents and index. Best of all, Index is nearly free of charge because it was placed in the public domain. If you need a more complex table of contents and a multilevel index there is Documate Plus, which can be used on a long manual or a book. If it's going to be set in type and printed, you need to know just how long it is—in characters, not typewritten pages—and Muchtxt, another WordStar utility in the public domain, will do the counting for you.

WordStar's ultimate advantage over Scripsit is in the area of portability. Because many different microcomputers use the CP/M operating system, WordStar and its utility programs can be used on any of them. Scripsit, however, runs on only the three micros that use TRSDOS—Radio Shack's Models II, 12, and 16. I won't argue the merits of such an arrangement, but I do believe it's dangerous to be completely dependent on one manufacturer. Should the future bring a CP/M microcomputer that I consider a better word processing tool than my Model II, I'll be able to transfer my software to it. That's nice. It will save me hundreds of dollars.

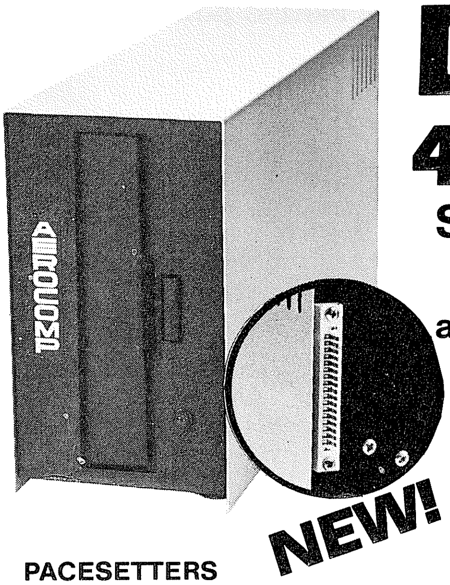
But when I talk about portability I'm not thinking primarily of the software. I have a vastly greater investment in the files I've produced over the years. Because they are ASCII files produced by WordStar, I'll be able to transfer them to a new system easily. Using Scripsit's ASCII conversion routine would make such mass transfer unthinkably laborious.

Portability is evidently a valuable consideration for numerous other people, too. Many microcomputer manufacturers either use CP/M instead of developing a proprietary disk operating system (e.g., TRSDOS), or offer CP/M as an official alternative to a proprietary DOS. Even Radio Shack, after several years of denying CP/M or CP/M software admission to its stores or its publications, will at last market a configuration of CP/M for its microcomputers. That is a tacit invitation to upgrade from Scripsit to WordStar.

The tendency among microcomputer people is to defend their systems against all others. I've used enough different computers and enough different software, especially word processing software, to know that everything is a compromise and that the "ideal" just doesn't exist. As far as the compromises that must be made right now, it seems to me that anyone with a Radio Shack business computer and the need to turn it into a serious word processing system would do better to scrap Scripsit and start using WordStar. The sooner the better.

Joseph Katz is a professor of English at the University of South Carolina. He is also a professional writer, editor, and consultant.

DISK DRIVES
40 & 80 TRACK
SINGLE & DOUBLE SIDED
as low as $169



PACESETTERS

Aerocomp leads the way to the BEST value in disk drives on the market. Quality, performance, reliability, warranty, service plus free trial – that's what you get from the leader. AEROCOMP!

BEST FEATURES

- ★ Fast 5 ms. track-track access
- ★ Single or double density
- ★ Easy entry door
- ★ "Flippy" feature allows read-write to the back of the diskette to cut media cost in half! (MPI)
- ★ Disk ejector (MPI)
- ★ External drive cable connection (no need to remove the cover to hook up the cable)

NEW!

MODEL III & 4 CONTROLLER and DRIVES

Convert cassette Model III or 4 to disk. Complete internal drive kits with 40 track SS drives (Tandon), Aerocomp disk controller board (will take up to 4 drives), power supply, mounting towers, all hardware & cables.

- ★ DRIVE KIT (no drives) $199
- ★ ONE DRIVE SYSTEM $369
- ★ TWO DRIVE SYSTEM $539

Shipping & handling $8.00 per system

- ★ AEROCOMP DISK CONTROLLER . . . $119

Shipping & handling $2.00 per controller

- ★ MOUNTING KIT w/o Drives & Controller . . $99

Shipping & handling $4.00 per kit

COMPLETE DRIVES

TRS80 Mod. I & III, IBM PC & TI 99/4A. Power supply & enclosure. Specify silver or almond. 5.25 inch.

- ★ 40 track single side (Tandon) $199
- ★ 40 track SS "Flippy" (MPI) $239
- ★ 40 track Dual Head (either) $279
- ★ 80 track SS (MPI) $299
- ★ 80 track SS "Flippy" (MPI) $329
- ★ 80 track Dual Head (Tandon) $379

Shipping & Handling $5.00 per drive

BARE DRIVES

Internal drives for TRS80 Mod. III, IBM PC, TI 99/4A, 5.25 in. (controller required)

- ★ 40 track Single Side (Tandon) $169
- ★ 40 track Dual Head (either) $249
- ★ 80 track SS (MPI) $269
- ★ 80 track Dual Head (Tandon) $339

Shipping & Handling $4.00 Per Drive

8 INCH DRIVES

Drive expansion box complete with power supply and fan. Tandon Slimline.

- ★ Two (2) 8" Single Side $699
- ★ Two (2) 8" Double Side $849
- ★ 8" Bare Slimline, SS $260
- ★ 8" Bare Slimline, DS $375

Shipping & Handling $5.00 Per Drive.

MODEL I STARTER PACKAGE

One 40 track SS drive, 2-drive cable, TRSDOS 2.3 disk & manual, freight & insurance (Tandon).

$249

MISCELLANEOUS GOODIES

- ★ TRSDOS 2.3 disk & manual $20
- ★ LDOS (Mod. I or III) $119
- ★ NEWDOS/80, 2.0 (Mod. I or III) $129
- ★ Diskettes (10 in library box) $23
- ★ 5.25" Drive Power Supply & case $59
- ★ 2-Drive Cable $24
- ★ 4-Drive Cable $34
- ★ Extender Cable $13

Shipping & Handling $2.00

PERSONAL CHECKS WELCOME

We'll be happy to accept your personal check with any mail order without any shipping delay.

FREE TRIAL OFFER

Use your AEROCOMP drive for up to 14 days. If you are not satisfied for ANY REASON (except misuse or improper handling), return in the original shipping container for a full purchase price refund. Applies to hardware only. Sorry, we cannot refund on software. We have confidence in our products and we know you will be satisfied.

WARRANTY

We offer a six months warranty on parts and labor against defects in materials and workmanship. In the event service becomes necessary for any reason, our service department is fast, friendly and cooperative. Our goal is 48 hour turnaround on all warranty or repair drives!

100% TESTED

AEROCOMP disk drives are 100% subjected to burn-in and bench test. We even enclose a copy of the test check list, signed by the test technician, with each drive. AEROCOMP means reliability!

ORDER NOW!

Order by mail or call TOLL FREE TO THE NUMBERS BELOW. Please note toll free lines will accept orders only. We accept VISA or MASTERCARD. Be sure to include card number and expiration date. We will not charge your card until the day we ship. Order by mail with credit card or send check or money order. Personal checks welcome. No delay. Order COD. No deposit required but all COD's will arrive cash, certified check or money order only. We'll send a card showing the exact COD amount before your shipment arrives. Shipping is **not** included in the prices shown. Texas residents add 5% sales tax.

CALL TOLL FREE FOR FAST SERVICE (800) 824-7888, OPERATOR 24

FOR VISA/MASTERCHARGE/C.O.D. ORDERS

California dial (800) 852-7777, Operator 24. Alaska and Hawaii dial (800) 824-7919, Operator 24.

TOLL FREE LINES WILL ACCEPT ORDERS ONLY!

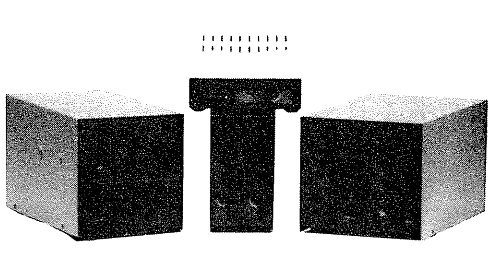
For information only call (214) 339-8324.
For TECHNICAL ASSISTANCE call (214) 337-4346.

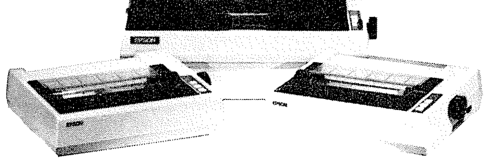
Dealer inquiries invited

AEROCOMP

Redbird Airport, Bldg. 8
P.O. Box 24829
Dallas, TX 75224

DISPLAYED VIDEO IS DRIVING DOWN PRICES ON DISKS! FREE SHIPPING!





TEC'S NEW HALF-HIGH 40 TRACK DISK DRIVE
W/CASE & POWER SUPPLY AT AN **INCREDIBLE** LOW PRICE! **$195.00!!!!!**
NEW! TRUE HALF HEIGHT DRIVES! 1 5/8 INCH NOT 2 INCH OR 2 7/16 INCH! DIRECT DRIVE! NO DRIVE BELT! 3MS TRACK TO TRACK!

TEAC's New Slimline 40 Track w/case & power supply	**$235.00**
Tandon 40 Track TM-100-1 w/case & power supply	**$220.00**
(DOUBLE SIDED 40 TRACK DRIVES ADD $75.00)	
Shugart New Slimline Double Sided 40 Track w/case & P.S.	**$299.00**
Shugart New Slimline Double Sided 80 Track w/case & P.S.	**$379.00**
Tandon Single Sided Slimline 8" Disk Drive w/dual case & P.S.	**$545.00**
Tandon Double Sided Slimline 8" Disk Drive w/dual case & P.S.	**$625.00**
Two Drive 5 1/4" Cable (FOR MOST COMPUTERS) (WITH GOLD PLATED CONNECTORS)	**$23.99**
DV's Color Computer 1st Drive Only	**$369.00**
ONE YEAR WARRANTY ON TEAC & SHUGART DRIVES/180 DAYS ON TEC & TANDON	
Hayes Smart Modem 300 Baud.......**$225.00** 1200 Baud	**$525.00**
Model I Double Density Board	**$89.00**
DV's Line Filter W/Surge Supression	**$45.00**
Diskette File Box "Holds 70 Diskettes"	**$29.95**

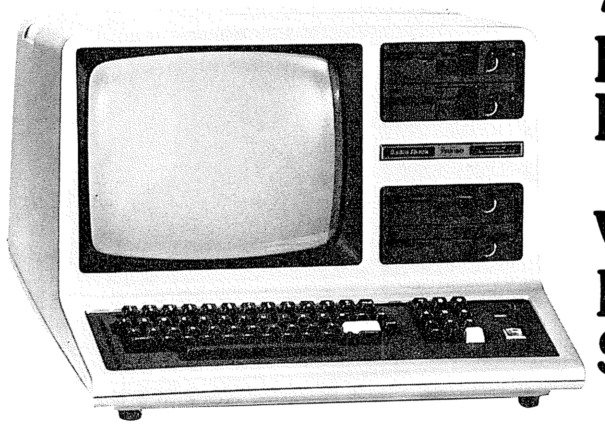
PRINTER PRICES

MX80 FT w/Graphtrax Plus.	**$459.00**	Gemini 10	**$349.00**
MX100 w/Graphtrax Plus	**$659.00**	Gemini 15	**$479.00**
RX80 w/Graphtrax Plus	**$389.00**	Citoh Prowriter 8510	**$395.00**
FX80 w/Graphtrax Plus	**$569.00**	Starwriter F10	**$1169.00**
FX100 w/Graphtrax Plus	**$779.00**	DWP 210	**$649.00**

Printer Cable 10' long w/gold plated connectors starting at **$25.99**

DISPLAYED VIDEO IS DRIVING DOWN PRICES ON DISKS

4 DRIVES!



4 DRIVES!

$1999.00 **$1999.00**

DISPLAYED VIDEO HAS DONE IT AGAIN!

THE STORAGE CAPACITY YOU'VE ALWAYS WANTED AT THE LOWEST PRICE EVERY CONCEIVED!
FOUR DRIVES IN A MODEL 4, 64K......**$1999.00**/128K......**$2079.00**
QUALITY BACKED UP BY DV'S SIX MONTH YOU CAN'T LOOSE WARRANTY!

DISPLAYED VIDEO is offering TRS-80' MODEL 4 disk drive systems fo **INCREDIBLY** low prices, quality backed up by DV's six month yo can't loose warranty.

MODEL 4 with 64K dual 40 track double density disk drives with TRSDOS 6.0 and TEN DISKETTES
PLUG IT IN AND GO......**$1599.00**/128K......**$1679.0**

MODEL 4 with 64K dual 40/40 track double density disk drives, wit TRSDOS 6.0 and TEN DISKETTES.
PLUG IT IN AND GO......**$1899.00**/128K......**$1979.0**

MODEL 4 with 64 dual 80 track double density disk drives, with TRSDOS 6.0 and TEN DISKETTES.
PLUG IT IN AND GO......**$1899.00**/128K......**$1979.0**

MODEL 4 with 64K dual 80/80 track double density internal disk drives with TRSDOS 6.0 and TEN DISKETTES.
PLUG IT IN AND GO......**$2199.00**/128K......**$2279.0**

MODEL 4 with 64K four internal drives of any configuration available t achieve up to 4 meg of disk storage	**CAL**
MODEL 12/1 Drive System	**$2695.0**
MODEL 12/2 Drive System	**$3349.0**
MODEL 16B/1 Drive System	**$4150.0**
MODEL 16B/2 Drives	**$5350.0**
MODEL 16B/Hard Drive	**$5850.0**
MODEL III/4 Internal Two Drive Kit: Includes controller board, dual drive mounting bracket, dual power supply, all hardware cable; and connector (gold plated) & TEC Drive	**$399.0**

4 DRIVE KIT MINUS DRIVE **$349.00**

MODEL 100 8K......**$679.00** 24K......**$839.0**

DV'S MODEL I DOUBLE DENSITY BOARDS **$89.0**

WE CARRY TRS-80, IBM, LNW, EPSON & FRANKLIN COMPUTERS

Visit our retail location at:
111 Marshall Street
Litchfield, MI 49252
(313) 426-5086
(313) 482-4424
(517) 542-3280
(517) 542-3939
(517) 542-3947

DISPLAYED VIDEO

IMMEDIATE DELIVERY
DV'S SIX MONTH
PARTS AND LABOR WARRANTY

Authorized Dealership at:
111 Marshall Street, Litchfield, Michigan 49252 **P142**
To Order: Call (313) 426-5086; (313) 482-4424; (517) 542-3280
(517) 542-3939; (517) 542-3947
OR WRITE:
Displayed Video, 111 Marshall St., Litchfield, MI 49252

CIRCLE NO. 13 ON INQUIRY CARD

Easy Menus

Now you can generate menus on disks quickly and easily, and with a minimum amount of typing.

by Andrew J. Kleinfeld

Menu-driven software can be a great help in simplifying the use of computers. This article describes a technique for generating menus on disks. The idea is that when you press <ENTER>, the computer will automatically list the programs, and when you press the number listed in the menu for the program, it will automatically run.

I developed this technique so that my secretary, then entirely unfamiliar with computers, could run the different programs we use in my law office without having to learn anything about syntax, Basic commands, or TRSDOS commands. She found it extremely helpful, and no longer had to refer constantly to her notebook of instructions.

This technique sacrifices elegance for simplicity, and minimizes typing. It is a technique, not a boilerplate program, so you will be reading this article and typing your own program, not copying the sample listings. I designed the technique so that I could remember how to do it and create menus from scratch on each disk, without having to refer to any paper. After the first one or two disks you set up this way, you probably

won't have to refer to this article.

First, create the screen display. Listing 1 is an example of a simple disk menu program. Lines 10-90 clear the screen and print the menu, skipping a line between each selection. Note that the last item is followed by a semi-colon, rather than a colon and another print statement. There is no reason to skip a line below the last item. The semi-colon prevents the computer from moving the cursor down a line, which will scroll the top of the menu off of the screen if it exceeds 16 lines.

Lines 100 and 110 scan the keyboard to find out which number the user presses. If the user accidently presses anything other than a number for a listed item, or has not yet pressed the number, the scanning continues in an indefinite loop. Once you press a number, the VAL(A$) function translates it from a string into a value. This translation is needed for the ON... GOTO statement in line 120.

Line 120 makes the computer branch to the appropriate command for running the selected program. In an ON... GOTO statement, the first line number will be selected if the value is 1, the second if the value is 2, the third if the value is 3, and so forth.

Lines 130-200 automatically load and run the selected programs. In so doing, they delete the menu program from active memory. The practical consequence for you when you are typing in the menu is that you should save it to disk before using it. When you use it, the program it loads will erase it. First save it to disk, then try it out and debug it if there are any problems.

If you want fancier graphics, they can easily be added with a subroutine. I omitted them to make typing easy and short. As prepared in listing 1, the structure of the menu program is so simple that after doing one or two, you will not have to load and modify an existing menu, or look at a listing. It is easy to write the menus from scratch.

Once you have saved the menu to disk, it is convenient to make it run automatically when you press <RESET>. To avoid having to check the directory repeatedly, I always call my menu MENU/BAS.

Go into TRSDOS to create your automatic menu run feature. In TRSDOS, type AUTO BASIC MENU/BAS. Once you have typed this command and pressed <ENTER>, the disk will always load and run Basic, and load and run the menu program when you press <RESET>. If you prefer not to run the menu in a particular situation, just depress <HOLD> as you enter date and time after a <RESET>. This will boot TRSDOS without executing the command to load Basic and run MENUS/BAS.

If you want to get back to TRSDOS when the menu is on the screen, press <BREAK>, then type SYSTEM. This is a "soft boot" of TRSDOS, simply restoring you to the TRSDOS previously loaded in. That makes it faster than pressing <RESET> to load TRSDOS. The soft boot should not be used when you change disks, since you may have used different versions of TRSDOS, either because of patches you put on some disks but not others, or because some programs use different versions of TRSDOS.

The menu program is easily altered to run machine language rather than Basic programs. Machine language programs are those that you run by typing their filenames while in TRSDOS rather than Basic, and pressing <ENTER>. Their suffix is usually /CMD, but this suffix need not be typed to run them in TRSDOS.

In order to run machine language programs from Basic, you use SYSTEM, followed by the filename in quotes. Listing 2 is a menu program that runs both Basic and machine language programs. This menu operates the disk for most of my law office daily computer operations. The Radio Shack Time Accounting and Profile programs are especially important to me. As you can see, the only modification needed is substitution of a SYSTEM statement for a RUN statement. Line 180 runs Profile by ordering the computer to do a SYSTEM "M". Line 190 similarly runs Time Accounting.

Incidentally, the extra INKEY$ in line 130 is not an error. INKEY$ flushes the keyboard, emptying the last key pressed into the string variable designated. Two INKEY$ commands in a row protect against accidentally or prematurely pressed keys. The first INKEY$ throws away whatever value is sitting there, in case it is garbage, and makes the keyboard report a null string to the second INKEY$. The second one gets the value intended from the user.

Things get a wee bit trickier for Basic programs that require memory size instructions or file instructions. As you know, unless the computer is otherwise advised, it overlays high memory machine language routines when Basic programs are run, and it doesn't reserve any buffers for file storage on the way to or from disk data files. Many graphics games have machine language routines for screen graphics or sound, requiring reservation of space. If you use these programs the hard way, you must look at a piece of paper everytime you use them to see what memory size number must be designated.

I find this an annoying nuisance, especially when the paper telling me what memory size to use is buried under a lot of other paper on my computer table. Here is how to avoid it.

Go into TRSDOS, and type BUILD followed by some filename. For the filename, use something which you will associate with the program to be loaded when you see it on a disk directory. For example, my Datestones of Ryn program requires me to set memory at 47778. I called my BUILD file for Datestones of Ryn, STONERUN.

After you type BUILD and the filename, the computer will give you instructions as you go. To limit memory occupied by Basic, the syntax is –M:memory size. Thus for Datestones of Ryn, my BUILD file consists of a single statement, BASIC STONES –M:47778. This is the same statement that TRSDOS uses to automatically load Basic and run a Basic program, with the addition of the memory size limiter. After typing your statement, type <BREAK>. The computer will automatically load the statement onto disk and call it a file with the suffix /BLD.

Then in your menu program, type SYSTEM "DO (name of BUILD file)" Line 130 of listing 1 is an example of this for my Datestones game. You should not type the suffix /BLD. DO is TRSDOS command that makes the computer run a BUILD file, with the suffix BLD. When the computer reads DO, it searches only for files suffixed BLD.

If you select the menu item requiring the memory size limitation, the computer will go to TRSDOS, reload Basic, automatically limit memory size, and run the Basic program. You never need to look up the memory size again.

File buffers for data transfers between memory and disk are usually specified as none when you enter Basic. You can specify any number from 0-15 buffers in response. If you simply press <ENTER>, the computer doesn't reserve any buffers. Some data file programs require that a different number of buffers be specified. It is as much of a nuisance to remember to do this as it is to remember to provide a memory size limitation. The nuisance can be similarly avoided by means of the menu program, in conjunction with a BUILD file.

Create a BUILD file just like the one

used to limit memory size. Instead of using –M:memory size, you use –F:number of buffers. For example, if a program requires that you reserve one buffer rather than the default setting of no buffers, your BUILD file would say BASIC (filename)/BAS –F:1. you can combine a memory size limitation and a buffer specification in the same statement. For a hypothetical program called DATAFILE/BAS, the syntax would be BASIC DATAFILE/BAS –F:1 –M:47000.

I like to write my menu program as I fill up the disk, starting as soon as two or three programs are saved. It is very easy to add to it. Here is the procedure when you have saved a new program onto the disk and want to enlarge the menu.

After you press <RESET> so that the menu loads, press <BREAK> to avoid operating it. List it out and use <HOLD> to stop the listing before the PRINT TAB(10) statements are off the screen. Add another one for the new program, using a line number higher than the last PRINT TAB statement and lower than the INKEY$ statement. If the line numbers are getting crammed too close together, type RENUM <ENTER> to make them increments of 10. This will automatically adjust the ON. . . GOTO statement appropriately.

The next step is to list again, and press <HOLD> or <BREAK> when your ON. . . GOTO statement appears. Edit the line by typing EDIT followed by the line number. Type X, which saves everything on the line, but allows you to add to the end. At the end of the line, type a comma and a line number higher than the last previous one. For example, if the last line specified in your ON. . . GOTO statement was 220, you would type ,230. Then save your altered menu by typing SAVE "MENU/BAS". Be sure to save it before using it.

Things get a little trickier if you have more than nine items on the menu. The easiest way to deal with this is to substitute an INPUT statement for an INKEY$ statement. INKEY$ responds to the first key you type, so it is most easily used for one digit numbers. INPUT just sits there until you hit <ENTER>, so it is easily used for varying numbers of digits.

If you really want to keep the INKEY$ function, you can do it by concatenating a string built by repeated INKEY$ commands and using a time delay. The idea here is that after you do an INKEY $ function, instead of dropping right down to the ON. . . GOTO statement, you go into a time delay loop designed to last for a couple of seconds. Another INKEY$ scans the keyboard during the time delay. If you type in a second digit, it concatenates it with the first digit. This is possible because both are stored as strings rather than values. Only after the time delay and concatenation do you insert the statement to find the value of the string and drop to the ON. . . GOTO command.

This seems like more trouble than it's worth to me, since nine programs on a disk are usually plenty. If you do pack your disks more fully, you will also need to print the menu differently on the screen so that it will fit without scrolling. One way to do this is with short menus, so that you can have two columns. You might have no tabbing for the first column, and a TAB(32) for the second column. Another way is to avoid skipping lines. Either makes the menu less attractive and slower to use. If your menus are densely packed, you will have more errors in which a person accidently presses the wrong number.

There is an additional little gimmick you can add to this program. Some of the machine language programs I have do not clear the screen before they run. Some for example, leave the TRSDOS information at the top of the screen. I dealt with this by inserting CLS: before my run statement in the menu. The effect is that when the user presses the menu number for the program, the ON. . . GOTO statement drops down to the correct line. That line first clears the screen, and then issues the command to TRSDOS to run the program.

You can also use this technique to add some instructions on the screen before the program runs. I don't like searching through my messy computer desk for documentation, so I make the computer give me instructions at the beginning of a program. Sometimes it is hard to do this in the program itself, because it may have such a complex network of machine language GOSUBs and GOTOs that it is hard to figure out where it really starts and where to put the instructions.

It is a cinch to put these types of program-specific instructions in the menu program. The RUN line can be turned into a GOSUB, the instructions PRINTed by the subroutine, and the RUN statement put in following the RETURN. A "Press Any Key When Ready" or "Press <ENTER> When Ready" INKEY$ technique can be used to run the program after the user has read the instructions.

Andrew Kleinfeld is an attorney practicing in Fairbanks, AK. He graduated from Harvard Law School in 1969. He uses his TRS-80s for various applications in his law office.

(Program listing on page 130.)

CIRCLE NO. 46 ON INQUIRY CARD

Snow Command: The Gameplaying Algorithms

by Patrick O'Connor

In the past two articles of this series, we looked at the "building" of a game program. In the first installment, we described the development of a storyboard, and last month we developed the flowchart and some of the "game environment" routines. These were routines in which we defined the actions of "nature" that did not interact with the user, but took place regardless of what the user did (the weather and the passage of time). There is another class of routines that do interact with the user and define how the computer plays its side of the game. These routines are the replenishment subroutine, the budget report subroutine, the inventory report subroutine, the weather report subroutine, and the street report subroutine.

There are also "gameplaying algorithms" that define some of the rules by which *you* play your end of the game with the computer. They are the routines for sending out crews on snowplows, sending out crews on salt trucks, sending out crews on garbage pickup, and killing two hours by taking a snooze.

We will develop these routines this month, and next month, we will see how all the pieces are tied together.

Note: Last month, I promised to use generally available Basic commands that could run on all TRS-80 machines. Since I am actually writing this program in Level II Basic, I can't promise that everything here will run on anything but TRS-80 Models I/III and the CoCo. Modifications needed to make this program run on the Model 100 should be quite minor, though. I have tried to modify some of last month's routines to make them more generally useful to owners of other machines.

This month, we'll write subroutines that interact with the player and define how the computer plays its side of the game.

If we take the routines we have to develop in order, the first is the one called the replenishment routine. What is a "replenishment"? In this case, it's the subroutine that allows you to purchase things. It is tied in closely with two other subroutines, the budget report subroutine and the inventory report subroutine. Whenever one of these two reports shows that something is amiss, you will have to take some action to correct the problem.

Suppose you are looking at the inventory report, and you see that the fuel for snow plows is running out. You will have to purchase more fuel. If you purchase too much fuel, the budget will be zero (or less!) when you look at the budget report. Then you will have to go back to the purchasing routine and say "Oops! That wasn't the amount I wanted to buy," which you can do by issuing another purchase order for a more reasonable amount of fuel. If you don't look at the budget report, you won't see that the fuel will bankrupt your department, and you will go broke when the purchase order "goes through."

Looking around at reports, (e.g., the weather report, budget report, inventory report, or street report) has no permanent effect on anything in the budget or inventory, so it's a good idea to keep in touch with things by checking these reports as often as possible. The purchase orders do not "go through" until you do something substantial, such as ordering out the plows or salt trucks, sending out a garbage collection crew, or just taking a nap for two hours.

You should be able to reach the purchasing (replenishment) subroutine whenever you see a problem that must be corrected by buying something. You would probably become aware of this kind of problem while looking at the budget report or the inventory report, so we arranged those reports in such a way as to allow you to buy something after looking at the report.

Last month, when we developed the weather mini-program, we found that it needs two parts. The first is the part that reports what the weather is like. The second is the part that figures what the weather will be like the next time we advance the clock two hours.

The inventory report, budget report, and purchasing report, like the weather report, require another subroutine—an accounting routine to keep track of purchases, revenue, and consumption. This budget, bookkeeping, and accounting routine keeps track of the goings and comings of salt, fuel, and

money. Like the weather emulator program, it figures out what everything will be like after a two-hour period has elapsed (see subroutine 5000).

Since it's hard to tell the players without a scorecard in this program, figure 1 provides a list of the important variables used in the budget, bookkeeping, and accounting routine.

What can happen to the budget and inventory in two hours? First, your department receives a certain amount of money from taxes, parking fines, parking meters and so on, in the amount of about $50,000 every two hours (line 5010).

Salaries paid to truck crews (line 5020) remove money from the treasury at a rate that depends on three things: 1) how many four-man crews are out on snow plowing or salting duty (each member of a four-man crew gets $20 an hour, or $160 per truck per two-hour period); 2) how many trucks you have called out (NP + ST); 3) what the percentage of trucks working (PT), is determined by how many trucks are actually on the road with paid crews.

Salt consumption (line 5030) depends on two facts: first, that each salt truck consumes about two tons of salt per two-hour period (per Chicago Bureau of Streets and Sanitation); and second, the number of salt trucks working depends on the number that have been called out (ST) and what percent of those are working (PT).

Fuel for trucks (line 5040) is consumed according to the number of trucks called out (NP + ST) and the percent of those trucks that are working (PT). Fuel is used at the rate of 10 gallons per working truck in each two-hour period.

If the temperature (TP) is less than 10 degrees Fahrenheit, calcium chloride must be used for road salt instead of sodium chloride (ordinary salt). Line 5050 decides what type of salt is being consumed, and what amount is left after two hours.

About 1,000 permanent salaries of $12.50 per hour are paid to staff and administrative people, regardless of whether or not crews are working on the streets (line 5060).

The cost of repairing trucks (line 5070) is handled in one of two ways. If there are spare parts in the inventory, the cost of repairing non-working trucks is automatically deducted from the spare parts, and all the trucks are put into working condition. If a purchase order for parts has been placed, the cost of the parts is deducted from the treasury, taken off the cost of repairs, and added to the inventory of parts. Then, later, non-functioning trucks are repaired in proportion to how many were purchased (line 5090).

If fuel is purchased, its cost—$1.35 per gallon—is deducted from the treasury (line 5100).

Salt is purchased at $14 per ton for road salt (per Morton Salt Company, Chicago), and $20 per ton for calcium chloride. These purchases are deducted from the treasury, and the quantity of salt (in tons) is added to the inventory (lines 5110 and 5120).

Building costs, such as rents, mortgages, utilities, and janitors, are $1,000 for each two-hour period (line 5130).

Line 5140 checks to see if the city is bankrupt. If it is, the game ends *immediately*, and you go directly to the end-of-job routine and become an unemployed bureaucrat.

Salt, calcium chloride, and fuel are checked to see if they are running out (lines 5150, 5170, 5190). If so, it appears that the player hasn't been checking the inventory report, so a message is printed on the screen to alert the player that supplies may not last out the next two-hour period (lines 5160, 5180, 5200).

Once the purchases of salt, calcium chloride, fuel, and parts have been made (added to the inventory and deducted from the treasury) the purchase orders (PS, PC, PF, and PP) are reset to zero so that the purchase isn't repeated (line 5210).

Note: Looking at subroutine 5000, you see a variable called DT# on nine of the routine's 22 lines. This was called "DT" in last month's column, but when my daughter, playing the game, accumulated more than a million dollars in the treasury, she saw the number $1.004E + 06 appear in the budget report. She thought the number was $1.00 and wouldn't buy anything for fear of bankrupting the city (an automatic loss of game). Because of this innocent misinterpretation, she ended up with the streets 97 percent blocked by snow and no garbage pickup for 41 days before she called me for help—her department was in such serious trouble there was no way to recover, and the game was lost. I immediately changed "DT" to "DT#", so that the number she had would look like $1,004,000 instead of $1.004E + 06.

> The amount of snow that piles up on the streets determines what percentage of roads is open. Two things that affect the rate at which snow piles up during a storm are the rate of snowfall and the wind speed. Even when it is not snowing, blowing, and drifting snow can block streets.

As you play Snow Command, you need to keep constant watch on budget and inventory to answer questions like "Am I running low on supplies?" and "Do I have enough money to buy what I want?" The budget report subroutine and the inventory report subroutine will display information on the screen that should answer these questions.

It will also provide a "pipeline" to get you into the Replenishment routine, which makes it possible to buy things when you need them.

Subroutines 13000 and 14000 use the same variables described in the budget, bookkeeping, and accounting routine. In that sense, the reports are just a way to get the value of these items to appear on the screen. There are, however, a few additional items in the reports that are not found in subroutine 5000 (see figure 2).

The first four items in the variable list are included in the heading of each report so the player can check the day and time. They are provided by the timekeeping routine described last month. At the end of each report subroutine, the player is asked whether to go to the purchasing routine (see subroutine 9000) or return to the menu. If you recall the storyboard for Snow Command, the game begins with a menu. The budget and inventory reports are called from the menu by pressing a key. After looking at these reports, you may want to either go back to the menu to do something else, or purchase something you're running low on. Q1$ and Q2$ are used to handle this.

Finally, we get to the replenishment routine. This is the subroutine that makes it possible to purchase needed fuel, salt, calcium chloride and repair supplies.

Actually, with the bookkeeping core (subroutine 5000) already provided, the purchasing routine is quite simple. All that has to be done to purchase salt, calcium chloride, or fuel is to

enter the quantity you want. INPUT statements store the numbers in PS, PC, or PF. When fuel is purchased, you are also appraised of the condition of repair costs. If you have more spare parts than you need to handle currently needed repairs, the cost of repairs shown on the screen will be negative, and you will not be asked if you want to purchase any parts.

If you have repair costs for which there are not sufficient parts, you will be asked how much to spend on repair parts. You can, if you want, spend more than the cost of repairs at these times. For instance, if you spend $2,400 when the cost of repairs is only $400, you will find that a stock of $2,000 worth of spare parts is now in the inventory. Repairs will be made automatically out of this stock of parts, and you won't have to worry about anything staying broken-down for more than two hours before it is automatically repaired, at least until the spare parts run out.

Last month, we had most of the weather report subroutine developed as part of a mini-program that displayed weather throughout the course of the winter. The weather-generating algorithm now has to attend to a number of other subroutines, providing information such as what type of salt to use (if the temperature is too low) and the rate at which trucks break down (if the temperature is low enough to freeze gas lines). With all those extra mouths to feed, you might expect the weather-emulating routine to be a lot more complicated than it was last month (see subroutine 4000).

Surprise! Nothing has been added. We wrote everything that was needed last month. The Weather Report, however, is another story. The PRINT@ statements in last month's report are unique to a Model I/III screen. To make things more generally useful, we have a routine without the PRINT@ statement, which you can adapt to fit the screens of other machines. You can "pad out" the spacing to center the lines for your screen whether it is a Model I, CoCo, or Model 100 (see subroutine 11000).

Note: In line 11060, GOTO 19040 returns from this subroutine to the menu after waiting for you to press the spacebar.

Since this is a subroutine now, it is called from and returns to the main menu.

The street report is a tabulation of how well you're doing at keeping the streets clear and ice-free. It is organized like the budget and inventory reports, and like them, the street report needs a "street condition" routine to keep track of what happens on the street after each two-hour interval (see subroutine 6000).

The amount of snow that piles up on the streets determines what percentage of roads (PR) is open (line 6010). Two things that affect the rate at which snow piles up during a storm are the rate of snowfall (RS) and the wind speed (WS). Even when it is not snowing, blowing, and drifting snow can block streets, depending on the height of drifts (HD) of snow already on the ground and the wind speed (WS) that blows the snow onto the streets. The effect of a snowstorm is more significant than that of blowing and drifting snow without any new snowfall.

It is also in line 6010 that the effects of snowplows and salting trucks in clearing roads is calculated. The number of

roads open is increased by an amount that depends on the number of plows (NP) and salt trucks (ST) called out, and the percentage of trucks working (PT). Line 6020 prevents snowfall from closing more than 100 percent of the roads and snow removal from clearing more than 100 percent of the streets.

Trucks freeze up if the temperature falls below zero. The percent of trucks still working will depend on how far the temperature is below zero (line 6040). Line 6050 prevents the

In computer programming, the phrase "garbage collection" usually means something else, but in this case, we are sending out 50 garbage trucks, the same trucks that are used as snowplows for major snow removal.

percent of working trucks from falling below zero percent or rising above 100 percent when repairs are made.

Plows and salters also break down randomly for mechanical reasons. Once in every 10 hours or so (line 6060) between two percent and six percent on the trucks break down.

Line 6080 calculates the cost of repairing plows and salt trucks at a cost of $40 per truck per two-hour period. If there are spare parts on hand, most (but not all) of these repairs will be done two hours after the trucks broke down.

As with the weather routine, the street condition routine is called as a subroutine every two hours, and returns when the new status of things has been computed.

The street report subroutine itself uses information provided by subroutine 6000. In addition to the status of plows and salt trucks and what percent of the roads are open, the street

Name	Type	Description of Variable
TP	Single Precision	Temperature (Degrees Fahrenheit)
PT	Single Precision	Percentage of trucks functioning
DT#	Double Precision	Dollars in the Treasury
NP	Single Precision	Number of plows (includes garbage trucks equipped with plows)
ST	Single Precision	Number of salt trucks
SA	Single Precision	Amount of salt for road salting
CC	Single Precision	Calcium Chloride (salt for low temperature road salting)
SC	Single Precision	Salt consumed (any kind)
FT	Single Precision	Fuel for trucks
CR	Single Precision	Cost of repairs
PP	Single Precision	Purchases of parts
PF	Single Precision	Purchases of fuel
PS	Single Precision	Purchase of salt
PC	Single Precision	Purchases of calcium chloride

Figure 1. *Variables for budget, bookkeeping, and accounting routine.*

report will also report on the status of garbage pickups, about which we will say more later (see subroutine 12000).

Note: In line 12110, GOTO 19040 returns from this subroutine to the menu after waiting for you to press the spacebar.

Now that we have completed all of the subroutines that define how the computer will play its end of the game, let's look at what you do to play your end of the game.

"Wait a minute," you say, "Didn't I play *my* end of the game when I purchased salt, fuel, or parts in the replenishment subroutine?"

Sorry, those orders aren't final until you do something "substantial."

To do something substantial, you have to take one of the following "actions":

- Sending out crews on snowplows
- Sending out crews on salt trucks
- Sending out crews on garbage pickup, or
- Killing two hours by taking a snooze.

You can call out snowplows two ways. Either you call out the standard crew of 50 snowplows, or you call out 100 trucks equipped with plows, including some garbage trucks that will stop picking up garbage to plow the snow.

Subroutines 15000 and 16000 simply put the numbers 50 or 100 into NP (the number of plows) and go to the timekeeping subroutine that figures out what happens during the next two hours. Both subroutines also include a mini-report that shows what effect your action has had on street blockage (a number on the screen that you can compare with what you saw in the last street report). As with other subroutines called from the menu, these end with a GOTO to the routine that waits for you to press the spacebar and then returns you to the menu.

As with the snowplows, salt trucks may be called out in two flavors—regular or extra salty (see subroutines 17000 and 18000).

In this case, the mini-report shows icing conditions on the streets that the trucks are supposed to de-ice. The numbers 50 or 100 are put into the variable ST (the number of salt trucks). Otherwise, these routines are identical to the snowplow routines.

In computer programming, the phrase "garbage collection" usually means something else, but in this case, we are sending out 50 garbage trucks, the same trucks that are used as snowplows for major snow removal (see subroutine 19000).

The amount of garbage picked up (G) depends on the percent of trucks on the street (PT/100), and what percent of the roads are open (PR/100). If all the garbage is picked up, then the trucks go back (NP = 0), otherwise 50*(PT/100) trucks are on the road (line 19010). Two hours' time elapses, and a mini-report on garbage pickup appears on the screen along with the updated time and date. At the end of this routine, lines 19040 and 19050 return to the menu when the spacebar is pressed. This is the GOTO that all the other routines used to return to the menu. (A name for this trick is "vectoring." It saves us the trouble of writing two identical lines at the end of each of the other subroutines.)

You may have noticed that there's nothing in this routine that creates the garbage our trucks pick up. Where does the garbage come from? The answer is subroutine 8000, the routine that takes care of what happens every two hours. That routine is called whenever time passes, and when time passes, whether there's a garbage pickup going on or not, garbage will pile up. That's why there isn't anything in subroutine 19000 that adds any garbage to the total. What else has been added to the timekeeping routine since last month?

Name	Type	Description of Variable
HR	Single Precision	Hour (time of day)
A$	ASCII string	Toggles between "am" and "pm"
P$	ASCII string	Toggles between "pm" and "am"
T$	ASC String	Temporary storage for toggling
M$	ASCII string	Month (Jan, Feb or Mar)
DA	Single Precision	Day of the month
Q1$	ASC String	Answer to: "Make any purchases?"
Q2$	ASC String	Answer to: "Return to Menu?"

Figure 2. Additional variables for budget report and inventory report subroutines.

In subroutine 8000, everything up to line 8050 is the same as it was last month. Lines 8060 to 8080 decide which player's turn it is and print the player's name on the screen. Garbage is produced at the rate of two percent every two hours in line 8090, then subroutines 5000, 4000, and 6000 are called to advance the budget, weather, and street condition two hours ahead. If garbage collection has not kept up with garbage production (line 8100), a warning message alert is flashed on the screen (see subroutine 21000).

If there is going to be a snowstorm (line 8110), a storm warning is displayed (see subroutine 22000).

That alert will only appear if a fairly large storm is expected. The prediction won't always exactly match the storm, but there *will* be a storm after each one.

If there isn't really anything substantial that you want to do—say, for instance, that there isn't any snow falling, the streets are 100 percent open, and 100 percent of the garbage pickup has been completed—maybe you'd like to take a nap on the job. This will kill two hours without actually doing anything. Don't do this when you're trying to plow snow, salt the streets, or pickup garbage, though. If you take a nap, the trucks will all come in, and the crews will probably all take naps, too, since the boss is asleep (see subroutine 20000).

After you enter the nap subroutine from the menu, you will see a message, "have a nice nap," on the screen. If you take a nap in the middle of a major blizzard, the message will be "how can you sleep at a time like this?" As with many of the other subroutines, this one vectors to 19050, which waits for you to press the spacebar, then returns you to the menu.

This completes all the subroutines you can reach from the menu. Next month, we will complete the game, and see if it can run on a Model 100 as well as Models I/III and the CoCo. We need to develop two more parts, the beginning of the game and its end.

Contributing editor Patrick O'Connor has held the position of professor of digital electronics at DeVry Institute of Technology for 16 years.

(Program listing on page 130.)

The new students at Dallas Baptist College have more access to a computer in one day than most students get in one year.

by Gary Gagliardi

STUDENTS

Key In on the Computer Revolution

In a recent article on computer use in education, *Time* magazine wonders, "What if all the students who could use a computer got access to one?" In American schools today, an estimated quarter-million computers are distributed among 44 million students. This means that if the use of these computers was distributed equally through the entire student population, each student could receive only 15 minutes of computer time per week, or less than nine hours of computer time per year.

On a hill overlooking a lake in suburban Southwest Dallas, Dallas Baptist College (DBC) has become a quiet beacon in the real computer revolution, an oasis of computer wealth in the vast desert of educational computer poverty. Many college administrators and academicians talk about how important computer literacy will be for their students' success in the future job market, but Dallas Baptist is one of the few schools that has acted in a dramatic and insightful way to ensure that their students are prepared for that future.

Beginning this year, *all* of the almost 500 students entering DBC for the first time—freshmen and transfer students—received their own personal computer. Perhaps even more significantly, the computers these students received were not locked in a laboratory somewhere or left to sit in the dormatory while the students attend class. Each student has a computer that can go where they go: a TRS-80 Model 100, the notebook-sized powerhouse from Tandy Corporation. This means that the new students at DBC have more access to a computer in one day than most students get in a year.

Like the majority of institutions of higher learning in this country, Dallas Baptist College is a small school. It offers 32 different majors leading to Bachelor degrees in the Arts, Science, Business Administration, Career Arts, and Music. Though it offers a major in computer science, its curriculum has no special accent on technical subjects. The administration of DBC is among the first to see the computer as more than a subject to be studied. They view the computer as a tool that should be used creatively to enhance all studies.

To most educators across the country, "computer literacy" seems to mean learning a few facts about the computer. In practice, this means that students spend a few hours listening to lectures on computer theory, and supplement this "knowledge" with even fewer hours at the terminal keyboard learning how to program.

But the person who coined the term "computer literacy," Berkeley computer educator Arthur Luehrmann, has a broader and more relevant definition of the term. He has defined computer literacy as "the ability to *do* computing and not merely to recognize, identify, or be aware of alleged facts about computing." This is the definition embraced by the program at DBC. At DBC computer literacy means accepting the computer as part of your everyday life.

To quote Dr. George Poynor, a professor of computer science at DBC who helps coordinate the faculty's implementation of the computer in their own individual classes, "If you are literate, it means you are able to use the computer

in a flexible, innovative sense. If the computer is not going to be something the student uses enough to become comfortable with, he isn't literate. The best way to implement computer literacy among students is to let them have the machine to balance their checkbooks, keep their little black books, and use for other out-of-class activities."

Dr. W. Marvin Webster, president of Dallas Baptist College, makes it clear how important he feels literacy is when he says, "Computer literacy is not an option for the educated man or woman of the 1980's. It's a necessity."

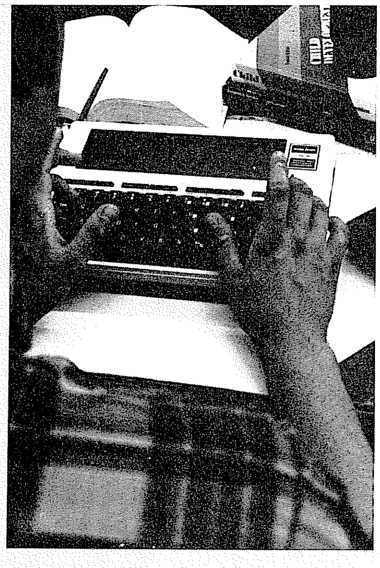
A large part of the credit for DBC's enlightened view of computer literacy must go to Bill Moos, who joined the college about two years ago as an assistant professor of computer science. He spent this summer with other members of the college evaluating computers that could be used for their computer literacy course.

After studying the use of computers by several other colleges, including the special programs developed by Drexel, Carnegie Melon, Case, and, especially, Clarkston College in New York, the new generation of lap-sized computers inspired the idea of using a computer that the students could keep with them as they attended their various classes.

Moos and the other faculty members reviewed several small, battery-powered computers that could be used effectively on a classroom desktop. Among the computers they evaluated were the Texas Instruments Compact Computer, the Epson HX-20, the Hewlett-Packard HP-85, and the TRS-80 Model 100. They settled on the Model 100 because it was user-friendly and because it included a built-in text processing program, as well as the Basic programming language.

Since they wanted to give the students as much memory as possible, they chose the fully-expanded 32 KB version of the Model 100. Initially, they only purchased about 30 of the machines to familiarize the faculty with the system before fall classes began. The new students didn't know that they would be receiving a computer until three days before registration.

To many of the new students, the fact that they were required to purchase a sophisticated computer came as quite a shock, but the lease-purchase plan developed by the college for the students helped cushion the blow. The computers —which would normally cost more than $1,100—were purchased at significant discounts through Radio Shack's educational division. The students pay a computer-use fee of $275 for the first semester when they receive their computers. This fee declines over the next three semesters, and includes both the computer's lease payment and the payments on an insurance policy protecting the computer against theft and damage. The computers belong to the students. They go with them when the students return home for summer vacation. At the end of four semesters, the students can pay $200 to buy out the lease and own their system outright.



Most students were very happy with the machine and are integrating it into their personal life as well as their school life.

The computer-use fee is required for all new students. For seniors transferring to Dallas Baptist College, who will not be attending long enough to pay off the lease, a special rental program is available. Returning DBC students can also take part in the computer leasing program if they so choose.

The students received their machines about a week after registration. Each is required to attend a computer literacy course three to four hours per week. Even though the Model 100 is very simple to operate by computer standards, the first weeks of this class had to deal with teaching elementary typing before students could get on to tackling more complicated subjects like Basic programming.

According to Dr. Poynor, "Some students have taken longer than others to master the machine. Some foreign students, for example, have never even seen a typewriter. We had to get everyone past that where-is-the-A-on-the-keyboard stage. It has taken about five weeks to get to beginning literacy. . . where an entire class can type in a line of programming without making a mistake."

After six weeks, the freshmen have taken to the computer so strongly that the clicking and beeping of computers in the classroom can become a distraction. Dr. Poynor makes the students cover the machines with a piece of paper to discourage them from playing with it while he is talking.

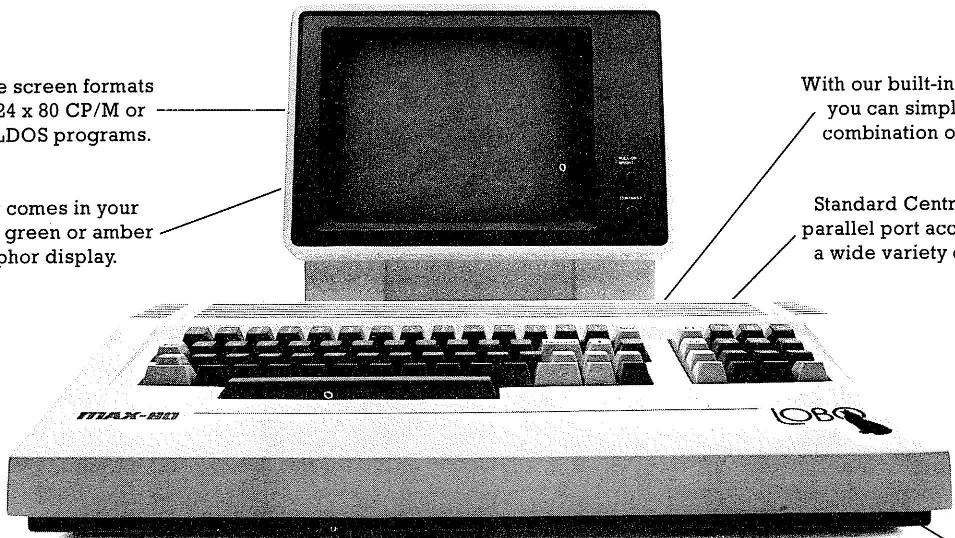
As the students have become more and more familiar with the machine, they are required to use it in courses other than computer literacy. English classes require that outlines and papers be prepared on the machine. Chemistry classes require that the students use the machine to calculate formulas. Psychology classes are planning to use the machine to analyze statistics. The goal is to find some use for the computer in *every* freshman course by 1985 and in every course in the school within three years. In working toward this goal, the faculty meets with Bill Moos for a half-hour every week to discuss their progress.

After six weeks, the machines seem to be surprisingly well accepted. A few faculty members felt as though they were being rushed into the program, but, in general, they are positive about this new tool in their classrooms.

Since the decision to buy the machines was made over the summer, many students were surprised when they discovered that they had to buy one. Some students thought that the machine was a waste of money and didn't feel as though it really applied to their course of study. A few students were opposed to the use of the computer in principle. But, according to those I talked to, most students were very happy with the machine, and, according to plan, are integrating it into their personal life as well as their school life. In school, students are using the computer to take notes and make outlines. Outside class, they are using them to store names and addresses, write letters home, and even to communicate with networks like Dow Jones and CompuServe.

After six weeks, there have been very few hardware problems with the computers. When I talked to Dr. Poynor, he was having a problem with a sticky <F4> key on his computer (the school allows faculty members to check out computers for their own use), but out of the almost 500 Model 100 computers

THE COMPUTER TANDY SHOULD HAVE BUILT.



Software-selectable screen formats let you run either 24 x 80 CP/M or 16 x 32 and 16 x 64 LDOS programs.

Monitor comes in your choice of green or amber phosphor display.

With our built-in disk interfaces, you can simply plug in any combination of peripherals.

Standard Centronics-type parallel port accommodates a wide variety of printers.

Our 5MHz Z-80B processor runs your programs 25% faster than TRS-80 Model 4.

Rugged construction includes heavy duty case and full-stroke sculptured keyboard.

Available storage includes Lobo's 5-1/4" floppy, 8" floppy and Winchester hard disks.

All Lobo hardware products carry a limited one-year parts and labor warranty.

AT THE PRICE TANDY SHOULD HAVE CHARGED.

The bottom line is this. For far less than the price of a TRS-80® Mod 4, you can own a lot more computer.

A computer that's 25% faster.

That supports both the CP/M® Plus and LDOS™ operating systems.

A computer that features a price tag of only $945 for the MAX-80® processor. And that expands with your needs without breaking your budget.

A FLEXIBLE SYSTEM AT A FLEXIBLE PRICE

When you put it all together —an entire system, including a dual 5-1/4" floppy drive subsystem (320 Kb of disk storage), monitor, CP/M and LDOS —totals up to just $1599.

Totally remarkable.

But that's not all. Because this basic MAX-80 system is incredibly flexible. Its dual operating systems run far more software than any other computer. (Including our specially-discounted Perfect-Calc™ and Perfect-Writer™ software packages.)

And since our MAX-80 features two RS-232C serial ports, you can easily add both printer and modem without switching back and forth.

SUPPORTS A WIDE VARIETY OF PERIPHERALS

What's more, Lobo offers almost any peripheral you might want to add to your MAX-80. At very special prices.

Like $995 for a 5Mb Winchester hard disk, the ultimate in fast, accurate data storage.

Or a whole line of affordable dot matrix and letter-quality printers. Or an 8" floppy drive. Or even a MAX-80 local area network.

EVERY PIECE OF HARDWARE BACKED BY OUR UNBEATABLE WARRANTY

Because every part of the MAX-80 system is so rugged and reliable, we go beyond industry-standard 90 day warranties. We back every piece of Lobo hardware with a full year warranty.

So now, you can run virtually all your TRS-80 programs faster and cheaper.

Just give us your Visa or Master Card number. We'll rush you the MAX-80. Try it out. Then, if for any reason you change your mind, return it within 30 days. We'll return all your money.

So call us. Because owning a MAX-80 costs you $945. But finding out about it costs you nothing.

1-800-235-1245

(1-800-322-6103 in California)

LOBO SYSTEMS, INC.

358 South Fairview Avenue,
Goleta, California 93117

CIRCLE NO. 27 ON INQUIRY CARD

BASIC DISK I/O

FASTER AND BETTER™

& OTHER MYSTERIES

New Book

IJG

CompuThoughts™

IJG is a small company. When we publish a book, it has to be excellent—"good" is not good enough.

We built our reputation by publishing what the other publishers would not. No "text book" or academic approach—our books are written in plain language—language that a novice or an expert can understand.

Lewis's new book, "BASIC Disk I/O Faster and Better"—the latest addition to our world famous "& Other Mysteries" series on the TRS-80—has no equal. It is also destined to become a best seller. His first was one of only four books out of over 400 to be rated at 100 in Nicita & Petrusha's "Reader's Guide to Micro Computer Books." Is this a great book? If it wasn't, I wouldn't publish it.

H.C. Pennington, President

430 pages
ISBN 0-936200-25-1

From Around the Globe, Users of Lewis Rosenfelder's First Book Said:
". . .the most useful book in my library." -Tom Prickett Elgin, IL ". . .a great book—everyone should have a copy." -E.A. Durso, M.D., Norman, OK ". . .excellent tips." Dr. Joseph Dillow, Vienna, Austria "It is worth far more than the purchase price." -Fred Hill, Savannah, GA "Wow! What a great book." -Gil Spencer, Spit Junction, Australia. Buy this book and speak for yourself—$29.95

IJG books are sold at booksellers and computer dealers world wide. If your dealer is out of stock, order direct. Include $4.00 for shipping and handling. Foreign residents include $11.00 in U.S. funds for shipping and handling. IJG, Inc., 1953 West 11th St. Upland, California 91786. (714) 946-5805

IJG

The Thoughtware Company™

CIRCLE NO. 23 ON INQUIRY CARD

purchased by the school, only four or five have had any problems.

Since the Model 100 is so easy to use, many students are surprised by how complicated other computers can be when they get a chance to use them in the school's computer laboratory. In this laboratory, the students get a change to experience IBM PCs and Apple IIe's. After using the simple and sophisticated Basic on the Model 100, doing Basic programming on an Apple IIe can be a real change.

The computer laboratory also gives the students access to printers to print out the work they type on the Model 100. The school purchased a dozen Radio Shack DMP-200s for this purpose. The laboratory is open from 8 am to 9 pm to give the students fairly convenient access to these peripherals. There is even some discussion of moving a couple of the printers to the dormatories so that the students can have even more access to them.

The lab also contains tape recorders for storing information on tape. (When the students received their machines, they didn't get tape recorders. The only peripherals they have are the acoustical coupler for telecommunications and the six-volt adapter.) The laboratory also contains a six-color, multi-pen plotter for those students who want to experiment with graphics output.

DBC is also installing a sophisticated Datapoint networking system to give the students the ability to use their Model 100s to communicate with the school's own network. This will enable them to store information generated by the portable Model 100s on the disks of a larger system. Unfortunately, the telephone company in Texas has gone on strike and delayed the installation of the 80 telephone lines that would give the students access to the larger network.

The computer lab is also used as a test site for Model 100 software available from outside sources. According to Bill Moos, having the laboratory enables them to control any outside software the school purchases for review. The students are not permitted to copy program tapes or photocopy the manuals. If any programs are found to be useful, they can be promoted through school newsletters and sold in the school's bookstore like any other educational aid.

If other peripherals (e.g., video screens and disk drives) become available for the Model 100, they will also become part of the computer laboratory. Bill Moos makes it clear that such Model 100 upgrades, though an intriguing possibility, are not required to justify the Model 100. The computers justify themselves easily with their current capabilities.

The only program that students currently felt a need for was a print-formatting program. When you print text stored in the compuer, the text-processing program prints only single-spaced. It will also print right off the end of the page if given a chance. After looking at one print-formatting program, they decided that it had too many features and used up too much memory for their purpose. To solve the problem, they are developing their own print-formatting program in Basic.

Strangely enough, the use of the Model 100 is seen by Dallas Baptist College as a movement back toward the fundamentals in education. In recent years, education professionals have begun to realize that, in many cases, the school system has failed to teach students such basics as English and mathematics. In this environment, computers may seem another distraction from those basics. But, in practice, the computer does not distract the student from these fundamentals; it concentrates the students' attention on them.

Dr. Poynor describes programming as forcing the student to be conscious of every character typed into the computer. This may be the first time that these students have ever been required to perform with such precision. If you type a character incorrectly, the program simply won't work. The student is exposed to a system in which incorrect spelling, syntax, and vocabulary are not forgiven. They are required to make the communication between user and computer work. The student learns the importance of a basic skill like correct spelling in a dramatic, visual, and effective way.

Similarly, without clear understanding of simple arithmetic, algebra, and Boolean logic, no student is going to be able to make his or her Basic programs work. Using the computer to solve problems in a program forces the student to master these basic skills. There are no shortcuts. If you don't master the basics, the system will never work.

The text processor, like the Basic language, is also seen as a tool toward a more complete understanding of language. According to Henry Becker, director of Johns Hopkins' classroom technology study: "One of the best uses for computers . . . would be to help teach kids to write." The cut and paste functions of a word processor take the drudgery out of revising a document. It frees the student from the non-productive task of retyping, so that he or she can concentrate on the more important fundamental task of getting things right. Professors can be more demanding of their students. They can make the student rewrite a paper without feeling that they are wasting the student's time by forcing them to spend a lot of time typing.

From this point of view, getting back to fundamentals, demanding better work from students, and the use of a computer go hand in hand.

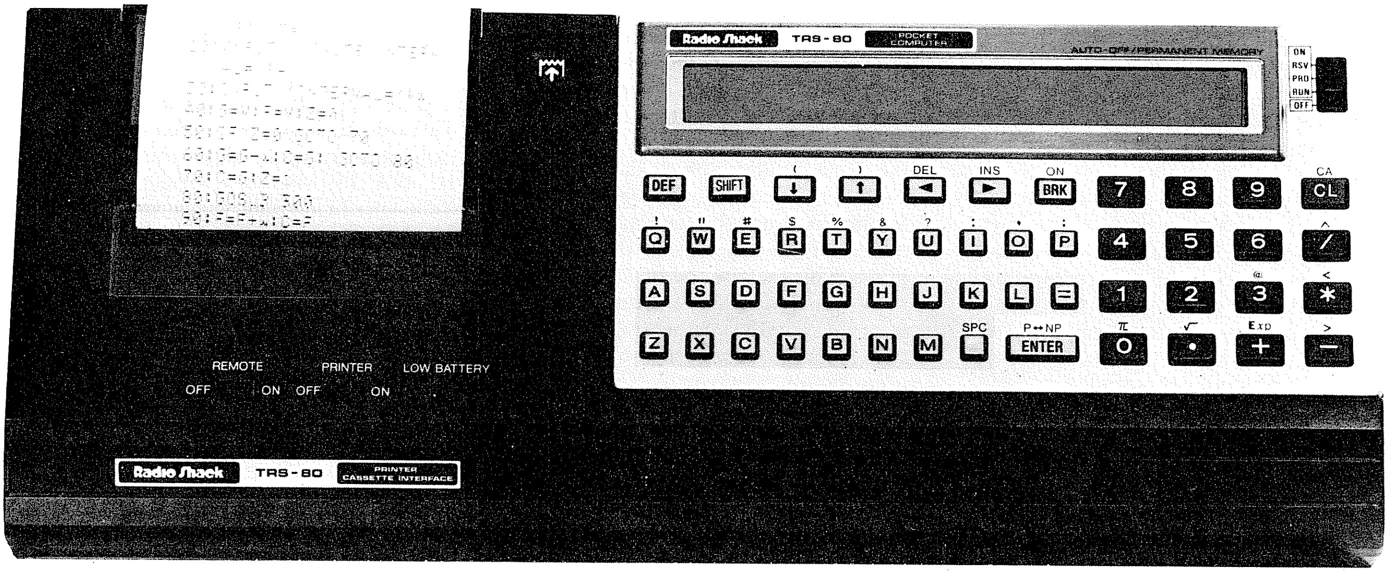
The *TIME* article about computers in education that I quoted at the beginning of this article ended with a rather grim evaluation: "The supposed computer revolution in schools seems barely underway." For America as a whole, this is certainly the case, but at Dallas Baptist, the revolution is in full swing. It may be too early to judge the effect on the students at DBC. It will certainly be years before similar programs are commonplace in other schools. But the first step is always the hardest and Dallas Baptist has taken that first step for us all.

Gary Gagliardi is the founder of Sof-Ta Consulting, an association of small computer experts headquartered in Seattle, WA. His first book, How to Make Your Small Computer Pay Off *(Wadworth Press), was published earlier this year.*

Polynomial Functions: A Different View

by Bob McElwain

This program makes the solution of any polynomial function a reasonable task by letting the computer do the tedious computation.



Few teachers would ask students to solve F(X) = 1.74X^5 − 1.92X^3 + 3.74X + 7.96 with pencil and paper. Too much time would be spent in meaningless computation. Yet it's misleading to allow students to continue to believe that Y = X^5 − 1 is a typical fifth order function. The approach offered here makes the solution of any polynomial function a reasonable task by letting a computer do the computation. Although written for the PC-3 and PC-4, the program can be easily implemented on any computer in any language.

It's possible to write (or acquire) programs that handle a variety of cases with ease. Enter coefficients; out come roots or zeros. However valuable such programs may be, students will gain little insight from them because the mechanics of solution are buried in the program used. The merit of the approach I'll describe here is that students view the entire procedure while leaving tedious computation to the computer. This approach also tends to reinforce the concept of approximation. Results used in the real world are generally approximate values.

In the programs provided, coefficients of a polynomial function are loaded, then a set of ordered pairs is generated by evaluating the function repeatedly. Although not required, it's recommended that students be asked to sketch the function, using the points generated by the computer. Given a sketch, it's usually easy to identify whole number bounds of roots or zeros.

The Solve routine requires entry of an initial point (X) at which the function is positive, and an increment or delta (usually plus or minus one), which when added to the initial point increases or decreases X in the direction of the root. The Solve routine increments X until the value of the function becomes negative. It then retrieves the last X for which the function was positive as an approximation of the root, divides the increment or delta by 10, and then continues iteration until the function once again becomes negative. Each approximation is closer to the actual root. The degree of accuracy is limited only by the accuracy of the computer.

Since problems of this type are not often presented in textbooks, they must be constructed. It's effective if every student has his or her own problem to solve. In constructing such problems, one can begin with something like:

Y = (X − 1)(X − 3)(X + 2)

which leads to:

Y = X^3 − 2X^2 − 5X + 6.

Adjusting the coefficients slightly forces roots or zeros that must be approximated in most cases. For example, changing 1 to 1.1, −2 to −1.9, −5 to −4.75, and 6 to 6.42 gives Y = 1.1X^3 − 1.9X^2 −

4.75 X + 6.42, which has approximate solutions of X = 1.1750, 2.5219, and − 1.9696. Adjusting the constant term only is often sufficient. Changing 6 in the initial function to 7 leads to Y = X^3 − 2X^2 − 5X + 7, which has approximate solutions of X = 1.1724, 2.8920, and − 2.0644. Any such adjustments can create a pair of imaginary roots. This is of little consequence if the degree of the polynomial function is four or more.

An advanced class can be asked to construct a set of functions for your use or their own. Give each student a set of integral roots from which a polynomial function is to be computed. Adjust each coefficient generated by a specific factor, perhaps 10 or 20 percent. Student constructions will contain errors and may contain more irrational roots, but if the functions are four degrees or greater, solution is still a considerable challenge.

The program uses a trick in evaluating the function Y = F(X) that allows the solution of any degree required. This can best be shown with an example. Suppose you have the following equation:

Y = A*X^3 + B*X^2 + C*X + D

In an unusual manner, the above can be factored as:

Y = ((A*X + B)*X + C)*X + D

This is a convenient form for computer computation, since no powers of X are directly computed. Further, if the coefficients are loaded into array A, the above can be written as:

Y = ((A(1)*X + A(2))*X + A(3))*X + A(4)

where A(1) = A, A(2) = B, A(3) = C, and A(4) = D, the constant term.

In the program, the subroutine, Evaluate Function, is derived from this last form. Let Y = A(1). Then if the following are computed in order, the last value of Y will be the value of the function for a given X.

Y = Y*X + A(2)
Y = Y*X + A(3)
Y = Y*X + A(4)

In Basic code, this might appear as

```
:Y = A(1)
:FOR T = 2 TO 4
:Y = Y*X + A(T)
:NEXT T
```

The Evaluate Function subroutine includes a further generalization. FOR Z = 2 TO W + 1, were W is the degree of the function, allows the evaluation of any function regardless of degree.

In preparation for use in the classroom, a set of functions for student solution needs to be available; usually this set must be constructed as described above. The computer program must be available, either loaded into a classroom computer or available to be loaded into students' individual computers. As listed, the program should load in any computer with Basic available, with only slight modification. For example, a DIM statement may be required for array A on some machines.

If only one or two computers are immediately available, it may be best to clearly define the task to be accomplished, then proceed with other classroom work, while allowing students to access a computer individually. Many students will require five or more minutes to enter coefficients and generate a set of ordered pairs, and all will need several minutes to generate roots. Thus this assignment can best be accomplished as a supplement to required work, with individuals taking time out as needed to use a computer.

Initial values and the direction in which to increment can be determined directly from the set of ordered pairs. However, requiring students to sketch the function (perhaps without graph paper!) has merit. Attention is focused on the approximate nature of the data and students may obtain a clearer picture of their task and the way in which the computer helps accomplish it.

In using the program as listed here for a pocket computer, note that nine decimal positions is a maximum, assuming a single digit integer portion of the root value. Iterations beyond this limit are meaningless on a machine with a maximum accuracy of 10 significant digits. Four to six significant digits offers reasonable accuracy for most situations. As written, an eighteenth order polynomial function can be dealt with. If a higher degree is to be investigated, a different data structure will be required.

Remember that the program counts the constant term as a coefficient. Be sure to enter zero values as appropriate. The increment must be plus or minus one for the count of required digits in Solve. Any initial increment will work, though; an approximate solution will be obtained.

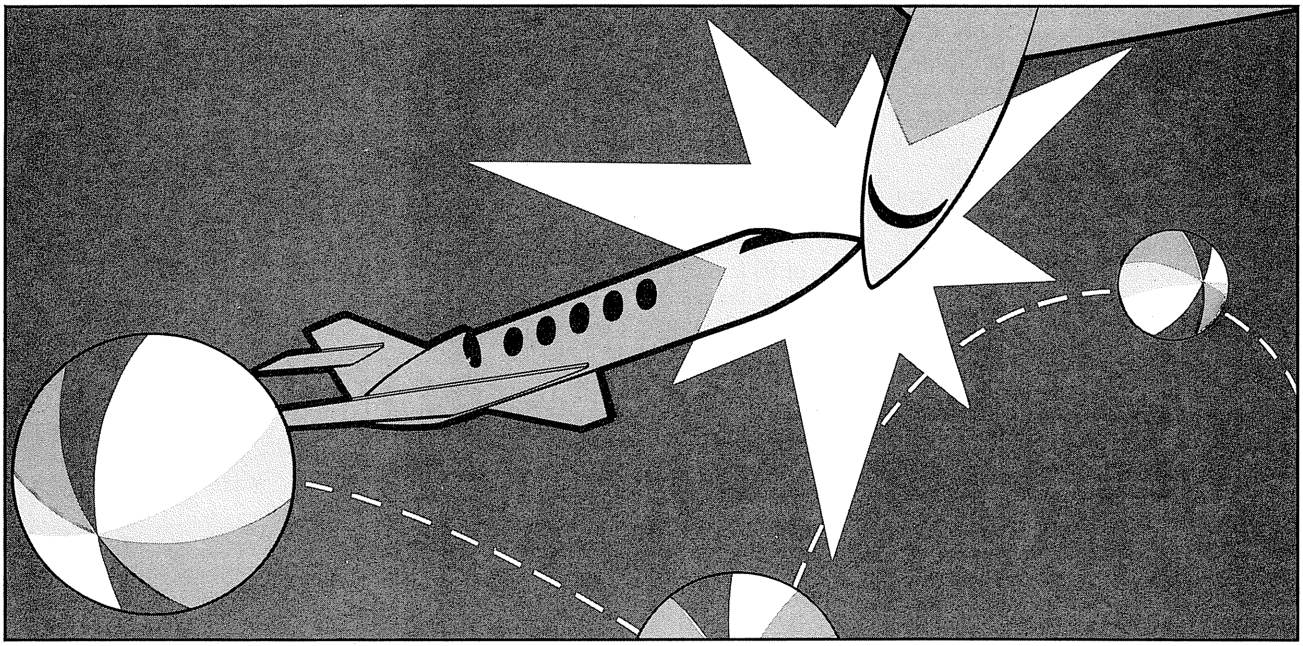
One measure of the degree of accuracy of the computed result is the value of the function at that point; it should be close to zero. If the root approximation is entered as the initial point in Solve and BREAK is pressed on the next prompt, the value of Y can be displayed manually to find how close to zero it is.

Contributing Editor Bob McElwain is a graduate of Pepperdine College with an MA in mathematics from Boston College. He is currently a computer science instructor and a software consultant for business and industry.

(Program listings on page 132.)

65024: PEEK The Magic Number

by Richard Ramella



Used as part of the PEEK command, the number 65024 creates character graphics on the Model 100 screen, and allows them to perform maneuvers in relation to each other's position, existence, and quality.

The number 65024 helps me fulfill a daily need to blow up rocket ships, crash airplanes, and help little robots find missing beachballs. I write computer games.

It's not documented in the Model 100 manual, but the number 65024 is the key to creating character graphics that interact in their liquid crystal world with a certain reality. Used as part of a PEEK command, this number gives heft, bounce, and even a simple intelligence to graphics characters. It allows them to maneuver in relation to each other's existence, quality and position.

For game people there lies ahead a tricky Model 100 arcade game called Cloud Rider. It's written in Basic and will easily fit an 8 KB machine. The prime purpose of its inclusion is to explain how PEEK graphic commands work.

The number 65024 refers to the first PEEK position on the LCD screen. Remember those five digits and we will shortly soar.

But first we must briskly walk through some fundamental ideas about the Model 100 screen and the graphics it displays.

Page 210 of the Model 100 manual features a video display worksheet. The top grid is the important one. It's the PRINT @ Worksheet, and it shows the screen's 320 PRINT @ positions, which read like the lines in a book, starting with 0 at the upper-left position and ending with 319 at the lower-right. There are eight lines of 40 positions each. Reading vertically down the left side of this grid, the PRINT @ positions of succeeding lines are in increments of 40: 0, 40, 80, 120, 160, 200, 240, and 280. It wouldn't hurt if you wrote these numbers appropriately leftward of the positions. Once you start using PRINT @ and

PEEK, you'll refer to them regularly.

The PRINT @ command is spectacularly under-documented in the manual. It isn't in the index. It isn't mentioned in the section on Basic keywords, though other more exotic PRINT statements are covered in pages 168-172. I finally found it on page 126, where it is given short shrift in this entry: "PRINT @ prints data at specified screen position."

That's not enough. To understand how to PEEK screen position values, it's necessary to learn how to put the values on the screen in the first place. The following is the correct form for printing the letter X at screen position 100:

```
PRINT @ 100,"X"
```

Any expression that works with a simple PRINT command also works with PRINT @. You may PRINT @ numbers (e.g., 1, 120, 4.45), defined variables (numerics such as A, B1, and strings such as A$, ZZ$, and V1$), and literals so long as letters are in quotes and numbers are not.

When you know how to PRINT @ on the screen, you have the first of two concepts that will unlock the secret of using PEEK to do interesting things with graphics. The short program Screen Peek Demo (see program listing 1) explains the PEEK command. Consider keying in that program before continuing.

Line 130 of Screen Peek Demo prints the word APE starting at PRINT @ position 1. Line 150 gives you a choice of which of the three letters you wish to check. Run the program and choose 1, which is A. Line 210 is an example of screen PEEKing: Z is made equal to PEEK(65024 + X), and the value of Z becomes the number value of the character. Now substitute the phrase "ASCII value" for "number value." Z equals the ASCII value of the character found at the screen position checked. Refer to the ASCII Character Code tables on pages 211-216 of the manual. Zero in on page 212, and go down the decimal column on the left of the page until you arrive at the number 65. In the third column to the right of that you will see the letter A. The number 65 is the ASCII value of the letter A. If you go to Basic command mode and type PRINT CHR$(65), the result will be an A. And if you run the program above and choose to check the first letter's screen position, you will discover that the PEEK command in the program recognizes the ASCII value at the PEEK position you told it to check. All the charcters in the ASCII tables can be similarly checked.

Now let's try a short program that makes a check of a similar, but different sort.

In Hubert Finds the Ball (see program listing 2) there is a hint of robotic intelligence, dim though it may be. Hubert is the temporary name I've given CHR$(147), which looks like a little person to me. Hubert appears at the top-left of the screen and invites you to hide a ball from him. You may put the ball at any position from 1 to 279. When you have made your choice, the program throws in 20 X's and hides the ball at the place you selected. Hubert starts at PRINT @ position 0 and makes a methodical search for the ball.

The interesting part of this simplistic search is this: With the PEEK command, Hubert is granted the "intelligence" to distinguish between a ball, a void, and what is neither ball nor void. If nothing is ahead, he keeps going. If he finds an X, he comments, "This is not a ball!" and continues. When he finds the ball, he announces his success.

Most of the Hubert program simply gives shape to the fantasy of the event that is dramatized. The important lines are 130 and 300-320. In line 130, PK is given the value of 65024, the magic number. In line 300, Q is given the PEEK value of PK (65024) plus J (the PRINT @ position) plus 2 (two positions to the right of Hubert's current position). Hubert is actually two characters wide. The first is a space and the second is Hubert, so it's necessary to check two over from the empty print position to discover what is immediately to the right of Hubert.

In this program, Q can only have three values: 32 for a space, 120 for the X, and 79 for the O.

If neither X nor O is encountered, the program goes to the next PRINT @ position. If Hubert encounters an X in line 310, the program goes to a subroutine (lines 340-410) in which he states that what he encountered is not a ball, jumps up and down a few times, and returns to the search. In line 320, if Hubert finds the ball, the program goes to its conclusion in lines 420-470. Here, Hubert announces success and generally dances about in puerile glee. Intelligence, he has some; decorum, he has none.

There are three tricky bits of screen PEEK information you should know. First, CHR$(32) is a blank and so is CHR$(224). The difference is that all blank spaces on the screen begin with ASCII values of 32. ASCII values of 224 must be assigned. If you PRINT one and PEEK its position for the other, it may seem like something is wrong unless you realize what is involved.

Second, if you PRINT @ 319, the final screen position, a line feed occurs, throwing all character graphics on the screen up one line. Tests directed toward their former positions are fruitless.

Third is a matter akin to having to deny a rumor. If you have worked with other TRS-80 models, you may know of the POKE command. POKE and PEEK are the salt and pepper of Basic. Screen POKEing does not work on the Model 100 in a useable way. Even so, let me briefly explain the relationship. POKE is another way of printing characters at specified screen positions. If it did work, typing the command POKE 65064,239 would put a black square at POKE-PEEK position 65064, which is PRINT @ position 40. To see what I'm talking about, try the following:

```
100 CLS
110 X = 65024 + RND(1)*319
120 POKE X,239
130 GOTO 110
```

When you run this short routine, nothing evident happens. But break into the program, type LIST, and tap ENTER. The program is listed over the vestiges of the POKE graphics. Since they won't show up during a program run, you must depend on PRINT @ to put graphics on the screen.

There are other valid Model 100 POKE commands for non-display purposes, but they have nothing to do with the topic here.

I think I've explained PEEK clearly enough that I can't be accused of giving you a broad general outline which you are expected to implement. So please accept the following suggestions as what they are—possible exercises in which you might broaden your ability to PEEK screen values.

1. Print CHR$(239) 35 times randomly within print positions 10 and 218. Establish Hubert at print position 0, then try to move him, using directional key taps, to position 319. If he comes in contact with a CHR$(239) graphic, the game is lost.

2. Print "1234567890" on the bottom screen line. Then establish a cursor somewhere above. Have the program randomly select one of the numbers below. Then, through the wonder of programming, let the cursor be moved by directional key taps and have the screen display RIGHT or WRONG, depending on whether the cursor comes to rest atop the requested number or not. What we have here is the kernel of a child's learning game in which the alphabet or another multiple-choice list could be substituted.

3. Frame the perimeter of the screen with CHRS$(239). Then send CHR$(255) off at a 45-degree angle with instructions to carom at 90 degrees from its current path any time it encounters something

other than a void on any of its sides. This is tricky.

4. Make the letter O into a ball and let it bounce rightward across the screen off a floor of CHR$(239) characters and in decreasing apogees. You may call this opus "The Apogee and the Ecstasy."

5. Create a game of tag in which you control Hubert's up, down, left, and right movements to elude an It which pursues him with single-minded purposefulness. If you are of a bellicose nature, change the elements into a jet fighter and a heat-seeking missile.

So much for broad general outlines. Now let's go to Cloud Rider and discover how it works, especially the PEEK statements.

The fantasy of this game is that you are a pilot delivering the mail through an increasingly violent storm. If you run into anything in the air, the game ends.

Let's play first and learn afterward.

When the program is keyed in, make sure the CAPS LOCK key is up, then type RUN and tap ENTER.

The screen will darken and in a window appears the message: "Cloud Rider must rev 1 minute. Patience." There's nothing to do but wait out the minute.

At the start of the action, you will see an airplane, CHR$(133), on the bottom line of the screen, and to its left will be an asterisk. Use the comma key to take the plane left, the period key to take it right. Your immediate goal is to make contact with the asterisk. This scores points, and it also causes the asterisk to switch its position to the other side of the flight path. Keep making contact with the asterisk to score points.

The situation is made quite perilous because objects begin to fly toward you. On the first pass, they are combinations of the four triangular shapes of CHR$(251)-CHR$(254). The longer you stay airborne, the thicker the air becomes with the potentially lethal objects. If you collide with one, the game ends.

To the left of the flight path are two numbers. The bottom one is the distance you have traveled and the top one is the score.

There is one more complicating factor. At the conclusion of each 67 kilometers, the storm intensifies with the addition of strings of CHR$(255) clouds in the sky ahead. The air is temporarily clear but becomes crowded again. You can't survive forever in this game.

Now let's look at the program in all its succinct glory.

Line 110 clears 2000 bytes for string space.

In line 120, A through C are defined as strings.

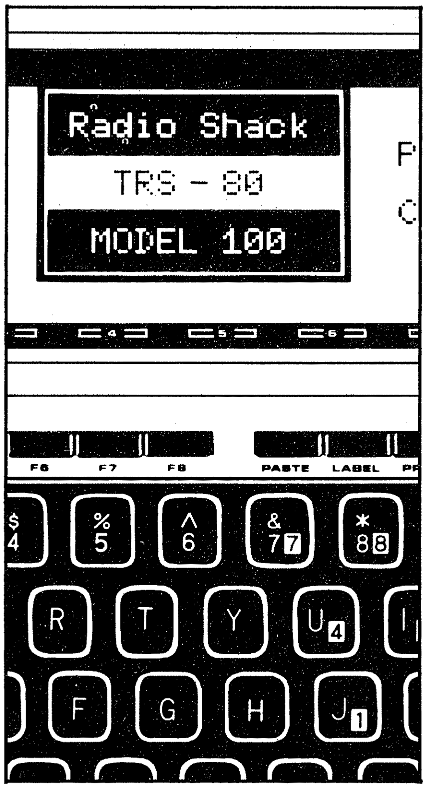
Lines 130-150 thwart the Model 100's proclivity for offering up the same series of "random" numbers every run of a program. Here, random values are used to an extent equal to the seconds reading in the current TIME$ value. This means the program can build a skyful of *objects d'crash* in 60 different ways.

In line 160, Q is made equal to the first PEEK position on the screen.

A, previously defined as a string, is made into the airplane, CHR$(133), in line 170.

In line 180, B, previously defined as a string, is dimensioned 75 deep.

The loop in lines 190-210 makes B(1) through B(75) into strings of 24 spaces,



CHR$(224). These are the voids into which the sky objects will later be randomly loaded.

Most of the screen is darkened by printing strings of 39 CHR$(239) in lines 220-240.

In lines 250-290, the one-minute wait message is printed. The wait is necessary because of the time required to randomly assign sky objects their places within B(1)-B(67).

Line 300 takes the program to a subroutine (lines 590-650) for assignment of sky objects.

The flight path is cleared by printing strings of blanks, CHR$(224), in lines 310-330.

Line 340 assigns W as the starting position of the plane, the PRINT @ position 292, which is located on the bottom screen line.

An asterisk is printed at the left of the flight path in line 350.

In line 360, Z is given the value 67 because the program will soon print, in succession, seven lines of the sky strings in the flight path. In its first printing, it actually prints B(67) through B(75), then is reduced by one each time it prints, creating the illusion that the sky objects are moving toward the plane.

Line 370 gives K a value of 10 as the starting position of the seven B(X) sky lines that will be printed. It is increased by 40 for each succeeding sky line printed so the sky and its objects appear and move regularly forward.

Line 380 begins a long loop that ends at line 480.

With each pass through the loop line 390 prints the Xth (1-7) line of the sky—B(X)—at the succeeding K positions.

Line 400 increases K by 40 to the next print position, just below the previous one on the screen.

An INKEY$ value is made possible in line 410, so certain actions can be taken if certain keys are tapped during execution of lines 430 and 440.

W, the current position of the airplane, is stored in S in line 420. This reduces plane flicker.

If the plane's position is not at the left of the flight path, and if C (the INKEY$ value), equals a move left command, then line 430 moves the new plane position to one left of the old, and the old position is erased.

If the plane's position is not at the right of the flight path, and if C equals a move right command, then line 440 performs a rightward action similar to that of line 430.

In line 450, the plane is printed at its current position.

If an asterisk is encountered on either side of the airplane, line 460 sends the program to a scoring subroutine in 730-810. This is tested with PEEK. Here Q equals 65024. In two PEEK statements, the position on either side of the plane is tested. If the ASCII value of 42 is found, the plane has contacted an asterisk and goes to the score subroutine.

Line 470 is a loser test. If any of the sky objects exists on the line above the airplane position—PEEK(Q + W – 40)—the program realizes it because it fails to find a space—CHR$(224). Accordingly, it goes to the loser routine at 660-710.

The sky printing sequence ends in line 480.

In lines 490-500, D (the distance traveled) is increased by one and printed at position 280 on the screen.

By decreasing Z by one in line 510, the program prepares to print the next series of sky lines, starting one earlier than before. It will reduce to 1 and start over at 67 without end until the game is lost.

If the course of sky lines is not exhausted (Z = 0), the program returns to print the next set of seven sky lines in line 520.

Line 530 flashes a warning that the storm is intensifying.

A loop beginning line at 540 throws more obstacles into the sky. A string of CHR$(255) from one to four characters wide is inserted approximately every two sky lines.

In line 550, N is given a random value of 1 or 2.

If N equals 1, line 560 loads the sky line with a CHR$(255) string placed randomly within it.

The loop ends in line 570.

Line 580 sends the program back to line 360 to start printing sky lines of the next series.

In line 590, the program has gone from line 300 to the initial sky obstacle-loading subroutine which continues through line 650. A loop begins, calling each line of the sky, B(1) to B(67).

A nested loop begins in line 600, calling each MID$ position within the current B(X) line.

J becomes the integer of Z/2 in line 610. Because J's chances of equalling 1 decrease as the value of Z increases, it is here that the increasingly crowded sky is created (see next comment).

In line 620, if J = 1 then one of the four character graphics from CHR$(251) to CHR$(254) is assigned a random position within the string.

The loops play themselves out in lines 630-650, and the program returns to start the game with all sky objects loaded into place. In all, 1,875 positions have been dealt with.

The loser routine begins in line 660. First, the distance figure is blacked over.

The words "Final Score" are printed above the score in lines 670-680.

Line 690 sounds a random tone.

A random character graphic in the CHR$(225)-CHR$(269) range is printed at the airplane position in line 700 to suggest wreckage.

RT is increased by one in line 710.

If RT is less than 30, line 720 returns the program to line 690, otherwise it ends.

The score-increasing routine begins at line 730 and ends in 810. SS, the score, increases by an amount equal to the number of kilometers traveled multiplied by two. Thus, quick travel between the two sides of the flight path increases the score more frequently.

Line 740 prints the score at the screen position 240.

A loop in lines 750-770 sounds a distinctive series of tones.

The PEEK test in line 780 determines if the asterisk is at the left of the flight path. If it isn't, the program goes to line 820 to do what is necessary when the asterisk is at the right.

If the program goes to line 790, the asterisk must be at the left, so lines 790-810 black over that position and an asterisk is printed at the right before the program returns to play.

The PEEK test of line 780 has determined that the asterisk is at the left, so lines 820-840 black over that position, an asterisk is printed at the left, and the program returns to play.

Line 850 is the END of the listing.

Richard Ramella has published more than 50 articles on computer-related topics. He currently writes for a California hospital.

(Program listing on page 135.)

A DISK ZAPPER for the CoCo

by Jeff Post

Written with Radio Shack's EDTASM, this Assembly language program will enable you to look at and modify disk sectors in hexadecimal or ASCII format.

In chapter 11 of the Color Computer Disk System User's Manual, Radio Shack has provided a wealth of information on directory structure for the Assembly language programmer. They were also good enough to provide us with the DSKCON routine for reading and writing sectors directly. Unfortunately, they neglected to tell us anything about the structure of files so that we can effectively use DSKCON. What we need to obtain the necessary information is a disk zapper program that will allow us to look at different types of files and find out how they are constructed.

CCZAP is an Assembly language program written with Radio Shack's EDTASM+. It will enable us to look at an modify disk sectors in hexadecimal or ASCII format. We can also make a printed copy of a sector's contents in either format. But beware! Romping about on the disk can be hazardous to its contents. Before writing a sector, make very sure that you know what you're doing.

The sector's data is read into a buffer that can be modified, and then is written back to the same sector or a different sector, allowing us to copy from one sector to another. The buffer is initially filled with random data (actually, whatever was in memory before CCZAP is executed, so it is important that a Read or Modify function to be done before writing to any sector.

To execute a command, press the key for that command, but do not press ENTER. The program accepts the following commands:

R reads the sector specified by the drive, track, and sector pointers. These points are initialized at drive 0, track 0, sector 1, but this sector is not read until the R command is given. The pointers can be changed with + (increment sector) or – (decrement sector) commands, or by entering new pointer numbers when asked by the R or W commands. This is done by entering one or two hexadecimal digits and then pressing ENTER. If you enter only one digit, then it becomes the second digit and the first digit defaults to zero (see figure 1). If you do not wish to change the pointers, pressing ENTER only will leave them unchanged. Pressing BREAK at any time will return to the menu.

W writes the current buffer contents to the drive, track, and sector specified by the pointers. The pointers can be changed the same as with the R command.

M modifies the buffer's contents. This command initially displays the sector's data in ASCII format, but the CLEAR key toggles the display between hexadecimal and ASCII formats. Pressing SHIFT and CLEAR together will dump the buffer's contents to the printer in either hexadecimal or ASCII, depending on which mode the screen is in. The track and sector numbers are also printed. The arrow keys will move the transparent cursor to select which byte to modify. If in ASCII mode, typing any character (other than the arrow keys, the CLEAR key, or the BREAK key) will put the ASCII code for that character in the buffer. If in hexadecimal mode, bytes may be changed by entering hexadecimal digits. Two digits must be entered for each each byte, and the first digit entered will go into the high nibble of the byte even if the cursor is on the low nibble to start with. When you've made all the changes you want to make, press the BREAK key to return to the menu and then use the W command to write the buffer to the disk.

B returns you to Disk Basic.

A Plus sign (+) increments the current sector pointer. If the pointer is on the last sector of the track, then the track pointer is incremented and the sector pointer is set to 01. Note that + means the key with the plus symbol on it; do not press SHIFT.

A minus sign (–) decrements the current sector pointer. If the pointer is on the first sector of the track, then the track pointer is decremented and the sector pointer is set to 12 (18 decimal).

The disk is organized into 35 tracks (0-22 hexadecimal), each of which is divided into 18 sectors (1-12 hexadecimal). One half of a track (9 sectors) is called a granule and is the minimum amount of space allocated to a file. Even if a file is only a few bytes long, it still gets a whole nine sectors all to itself.

When you have CCZAP entered and assembled, load and execute it and then read track 11, sector 3 (note that all


R.BURNS

track and sector numbers are in hexadecimal). This is done by pressing R. The cursor will move to the current drive number. Since we don't want to change it, just press ENTER and the cursor will then move to the track number. Enter the number 11 by pressing the 1 key twice, and then press ENTER. The sector number is changed to 01 by pressing the 1 key once followed by ENTER. As soon as ENTER is pressed,the sector will be read into the buffer. This is the first sector that contains directory entries. Each entry is 32 bytes long, and the first 11 bytes are the filename and extension for that file. The next byte is the file type (0 = Basic program, 1 = Basic data file, 2 = machine language file, and 3 = text editor source file). The 13th byte is an ASCII flag, and the following byte (the third from the right) is the one we are interested in. This byte is the number of the first granule allocated to the file and will be from 0 to 43 hexadecimal. To convert from the granule number to track and sector numbers, see figure 1.

Track 11 is the directory track and therefore does not have granule numbers assigned to it. If the granule number is even, the first sector of the granule is sector 01; if it is odd, then the first sector is sector 0A.

Find a machine language file in the directory (one with a BIN extension) and convert its granule number to track and sector numbers. Then read the sector specified. Examine the sector by using the M command, and press CLEAR to put the display in hexadecimal mode. The first byte in the first sector must be zero; it is a flag that tells the Disk Basic ROM that the data following is part of the file. The next two bytes are the length of the file. The fourth and fifth bytes specify the address where the file is to be loaded, and the following bytes are the data for that file.

The byte following the data will be either 00 or FF. If it is zero, then another block of data with the same format as the first will be loaded as part of the file. Because of this, machine language programs do not have to be loaded into a contiguous block of memory. We'll shortly see how this fact can be used to create a program that auto-starts (starts running as soon as it's loaded without the need of typing in EXEC). If the byte following the data is FF, then the end of the file has been reached except for four more bytes. The next two bytes are ignored by the Disk Basic ROM and the last two bytes are the EXEC address for the program.

Each granule is 9 sectors long, and each sector contains 256 bytes, for a total of 2,304 bytes per granule. What happens if our machine language program is

Gran #	Track #	Gran #	Track #
00, 01	00	22, 23	12
02, 03	01	24, 25	13
04, 05	02	26, 27	14
06, 07	03	28, 29	15
08, 09	04	2A, 2B	16
0A, 0B	05	2C, 2D	17
0C, 0D	06	2E, 2F	18
0E, 0F	07	30, 31	19
10, 11	08	32, 33	1A
12, 13	09	34, 35	1B
14, 15	0A	36, 37	1C
16, 17	0B	38, 39	1D
18, 19	0C	3A, 3B	1E
1A, 1B	0D	3C, 3D	1F
1C, 1D	0E	3E, 3F	20
1E, 1F	0F	40, 41	21
20, 21	10	42, 43	22

Figure 1. Conversions from granule numbers to track and sector numbers.

```
********************************
*CONVERT GRAN TO TRK & SEC*
*  On entry A register has gran #  *
*  On exit A = track, B = sector  *
********************************

         CLRB
         CMPA    #$22
         BLO     NOADJ
         INCA
         INCA
NOADJ    LSRA
         BCC     GRANO
         LDB     #9
GRANO    INCB
         RTS

********************************
*CONVERT TRK & SEC TO GRAN*
*  On entry A = track, B = sector  *
*  On exit A register has gran #  *
********************************

         CMPA    #$11
         BEQ     DIRTRK
         BLO     LOWER
         DECA
LOWER    LSLA
         CMPB    #$0A
         BLO     EVNGRN
         INCA
EVNGRN   RTS
DIRTRK   CLRA
         CLRB
         RTS
```

Figure 2. Two subroutines that convert granule numbers to track and sector numbers and vice versa.

longer than this? Where does the remainder of the data go? This is where the File Allocation Table (FAT) comes in. The FAT is stored in sector 02 of the directory track. Looking at the FAT we see that the first 68 bytes contain either FF, a number from 00 to 43, or a number from C1 to C9. The rest of the bytes in this sector (69-256) are not used and may contain anything (but usually not zeroes as the Disk System Manual claims). Each byte corresponds to one granule. If the byte is FF, it means that the granule is not used for any file. If it is C1-C9, it means that it is the last granule allocated to that file, and if we drop off the high nibble, the remaining number (1-9) is the number of sectors actually used by the file.

If the byte is 00-43, however, it means that more than one granule is used for the file and the number points to the next granule allocated to the file. With this method of allocating disk space it would be possible for one file to use up the entire disk! Of course, we don't have enough memory in the Color Computer to read in a file 156,672 bytes long, so eventually we'll point to a granule whose byte in the FAT will be C1-C9.

With all of this information, it should be fairly easy to write a program that can find the start, end, and execute addresses for machine language programs (except for the execute address, these are not stored in memory like they are for a program read in from tape). Or you can write your own program to dump data out to disk as a BIN extension file if you also remember to update the File Allocation Table. To do this you will need to convert granule numbers to track and sector numbers and vice versa (see figure 2).

Now, for that auto-start I promised. Implementing this function turns out to be very easy. Remember that the byte following the data is either 00 or FF, signifying more data to come, or end of file? Well, after the last block of program data and before the FF (end of file) byte, we insert an eight-byte block of data as follows: 00 to signify more data to load; 00 03 meaning a three-byte block; 01 82 as the address to load it at; and finally 7E XX YY, where XX YY is the execute address of our program. The byte 7E is the code for a JMP opcode, and 0182 is the address for the vector to Basic's line input routine. After the program is loaded with a LOADM function, Basic expects you to type in something and so goes to the vector at address 0182, which we have just intercepted. Simple, isn't it?

That pretty much covers the structure of machine language programs on disk, but what about Basic programs and data files? It turns out that these are much simpler than machine language files. There are no control bytes whatsoever, just ASCII data. (Basic program files contain tokens, however. Tokens are single bytes that represent Basic statements and will not be covered in detail here, since they deserve a separate article by themselves). The Basic ROM knows where the end of the file is from the last two bytes of the directory entry. These bytes define the number of bytes in use in the last sector of the file, and we have already seen that the last sector is specified in the File Allocation Table. Happy zapping!

Jeff Post is currently a systems programmer, developing software for Z-80-based systems. He has 17 years experience in analog and digital electronics.

(Program listing on page 137.)

GREAT PROGRAMS AT GREAT PRICES...

PROGRAMS SO GOOD
THAT WE GUARANTEE YOU'LL LIKE THEM!

We choose every program we sell based on quality, and for a very good reason; we GUARANTEE satisfaction with every sale. We take all the guesswork out of mail-order purchases by allowing YOU to be the judge of the product that you buy. Over the last year we've had an incredible 99.5% satisfaction rate, unmatched in the software industry. As well, every program you buy from us is fully supported by both the manufacturer and JMG Software. You can't lose!

SUPREME RULER and SUPREME RULER PLUS

(By JMG.) In this simulation/strategy game you become leader of a struggling country and must deal with your economy, your population, and your army. From 1 to 9 players can each take a country and compete, or you may have computer-controlled countries. A great game!
16K / 32K CASS $18.50 32K DISK $20.50.

(By JMG.) A 48K Extended version of SUPREME RULER. We could take up many pages describing it, but instead we'll quote one of our users: "I have hundreds of dollars in game software from Epyx, Avalon Hill, SSI, etc. RULER is the best of the lot. I love it!!" - G.C., Oneida, NY.
48K, either CASSETTE or DISK $26.50.

STARFIGHTER

(By Adventure International.) An incredible program that combines strategy with the realism of a spacecraft simulator. You get a sharp, out of the cockpit view, and you have very realistic controls at your command.
16K CASSETTE $24.50 32K DISK $26.50.

FLIGHT SIMULATOR

(By Sublogic.) This program turns your TRS-80 into a realistic flight simulator with an out-the-cockpit view of your surroundings. Gives an amazing 3-D view of your "world", with mountains at your side and a runway to land on, as well as other landmarks.
32K DISK $29.50

JMG IS ALSO THE PLACE FOR UTILITIES...

LDOS 5.1

(By LSI.) The number one operating system for TRS-80 Models I and III. Chosen by Radio Shack as a sophisticated alternative to TRSDOS, LDOS is loaded with features, from type-ahead to Model I & III compatibility of data disks, and much more. Supports Model I double density, and non-Radio Shack disk drives as well. Comes with an excellent manual.
Model I or III DISK $114.50.

EDAS 4.1

(By Misosys.) EDAS is the most advanced Editor/Assembler available for TRS-80s. It is a must for anyone doing serious assembly language work, and it is also ideal if you are just starting; as you become more proficient, you will find more use for the advanced features available. EDAS supports assembly from disk, macros, and MUCH more. Model I/III version requires LDOS; TRSDOS 6 for Model 4 version.
Model I/III (LDOS) or Mod 4 (TRSDOS6) $89.50.

SUPER UTILITY PLUS

(By Powersoft.) An extremely useful utility package that combines over 50 functions into one program. It supports almost every DOS, both Model I and III. With so many utilities combined into one in SUPER UTILITY PLUS, this program is a great value.
MODEL I/III DISK $64.50

DISASSEMBLER III

(By Misosys.) An advanced disassembler with many features and extensive capabilities such as direct disassembly from CMD files, full label generation, data "screening" for text areas, and more. Output to screen, printer, tape, or disk. Runs under most DOSes (incl. TRSDOS), on Models I, III and 4.
Model I, III and 4 DISK $34.50

TO ORDER: Please specify Model of computer, Disk or Cassette, amount of memory you have, program(s) wanted, and method of payment. We accept Check, Money Order, VISA, MasterCard, or COD. Please enclose $2.50 for shipping costs. We also accept orders by phone.

OTHER PROGRAMS AVAILABLE:

Z CHESS III	(Adventure Int'l)	DISK $26.50	CASS $24.50
GALACTIC TRILOGY	(Adventure Int'l)	DISK $34.50	
PROJECT OMEGA	(Adventure Int'l)	DISK $22.50	
CAPTAIN 747	(LSI)	DISK $27.50	CASS $29.50

OUR GUARANTEE:

We sell only top-quality software. If, however, you are unsatisfied with a product you have purchased from JMG, you may return it (in good condition) within 10 days for a refund, less a $2.50 handling charge for programs under $50 ($5.00 for programs over $50). We also ask that you send us a letter stating the reason for your return.

JMG SOFTWARE INTERNATIONAL

THE FINE PRINT:

Shipping Charges are $2.50 in North America, $5.00 overseas.
All prices in U.S. Funds.
Please allow 1 to 2 weeks for Personal Checks to clear.
Add $1.50 for COD Charges.
Model III programs will also run on Mod 4 in III mode.

P.O. BOX 598
FALLS STATION,
NIAGARA FALLS, N.Y.
U.S.A. 14303

OR

710 UPPER JAMES ST.
HAMILTON, ONTARIO
CANADA L9C 2Z8
(416) 389-6086

CIRCLE NO. 24 ON INQUIRY CARD

Biorhythms and Your CoCo

Whether scientifically proven or not, biorhythms can be fun to chart and interesting to study. Now your CoCo can help you in determining what your future moods will be like.

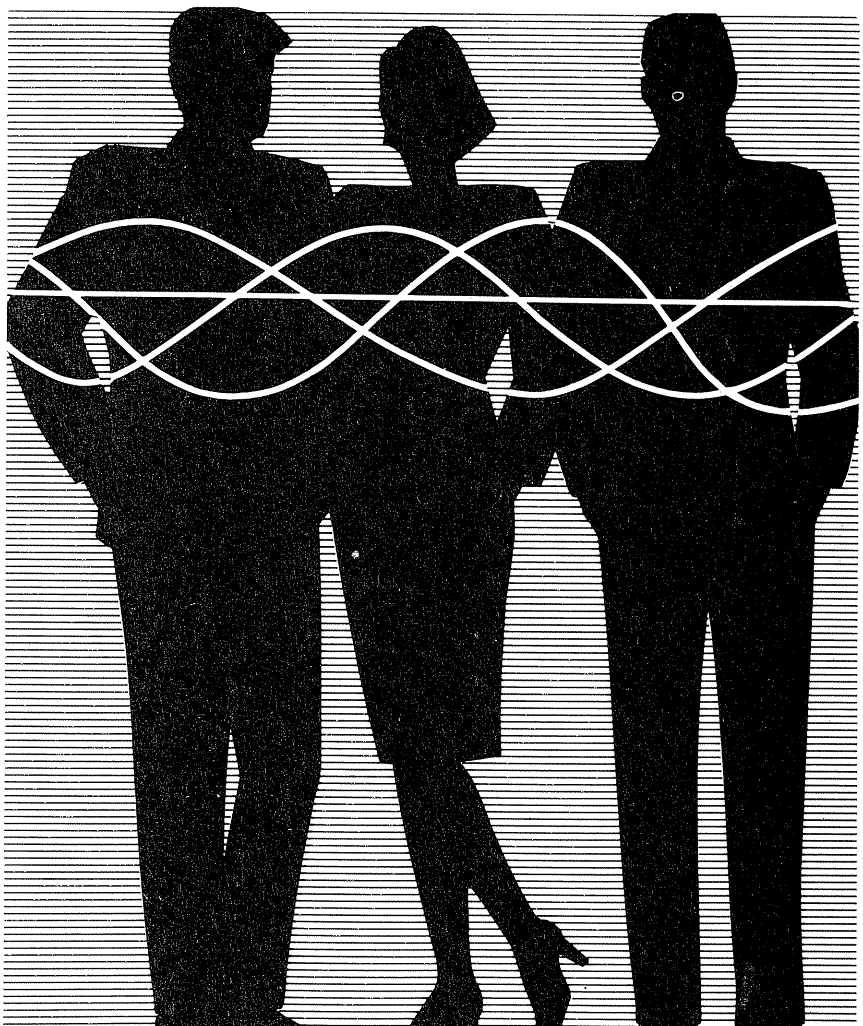
by Robert Toscani

Ever have days when, for no apparent reason, you just are not up to par? Wouldn't it be useful to know in advance when such days are likely to occur? Now, using this program, your handy, dandy, all-purpose Color Computer can gaze into your future and tell the days you are likely to be up or down. The program uses Color Basic and about 2.2 KB so it should run on a 4 KB CoCo and *may* even run on the MC-10. More about that later.

Before discussing the program, I would like to explain the theory of biorhythms on which it is based. I say theory, but from what I have read there seems to be no solid scientific evidence that biorhythms exist. Of course that never stopped anyone from believing whatever they wanted to. Besides, the program makes a pretty chart on the screen.

Biorhythms are three natural cycles that begin when a person is born. The cycles are labeled physical, emotional, and intellectual, and they last 23, 28, and 33 days, respectively. During the first half of each cycle, you are in its positive side and perform better. In the second half, you are in the cycle's negative or recharge side and do not do as well in that area. The worst time is usually considered to be when the cycle changes from positive to negative and vice versa. The days on which this happens are called critical days. The theory says you are at your lowest point on these days.

Of course you will rarely have everything happen together. For example, your physical might be positive, your emotional negative, and you are at a



critical day intellectually. No one cycle is considered all powerful, and each must be considered in relation to the others. In addition, the range of change is not constant, but varies from person to person. Anyway, the theory is that if you know when your cycles hit critical days, you can take more care on those days.

The old method of figuring your cycles involved a great deal of calculation. You also had to plot the cycles on a graph and produce a sine curve for each cycle. The computer will do all this for you, but due to the inherent limitations of the CoCo's low-resolution graphics, I decided to set up a second way to present the information.

The limitation is that although you have a 64-by-32 point screen for graphics, they are grouped into blocks of four points. If you set a color inside any block, the other points will take on that color if they are any color but black. To illustrate, set 0,0 to red. Now if you set 1,1 to blue, 0,0 will also turn blue. This will make any graph difficult to read, so I also provided a chart.

Perhaps if I run the program for you, you will understand. Upon start-up, you are presented with a choice of numbers or graphics. Pressing G produces the color graph, while N does the chart. This is an INPUT command so you must hit ENTER.

You will then be asked for a series of dates. They are, in order, your birth year, month, and day, and then the year and month you want to figure your cycle for. Use only numbers. You do not have to type in the 19 for the years questions unless you were born before 1901.

Assuming you had pressed N in response to the initial question, the screen would then display the month and year for the data about to be presented, a zero at the upper right corner, and three abbreviated column headings for the three cycles. Going down each column, you will see somewhere a plus or minus sign showing which phase the cycle is presently in. Then will come a number. This is a changeover date and thus is critical. All critical days for that month and each cycle are listed.

After absorbing that information, you will note the message Correction P/M? According to the theory, your cycles might be a day ahead or behind the computer's predictions, depending on the time of day or night that you were born. To adjust the cycles a day forward or back, press P or M, respectively. All future calculations will then have that correction, which will be indicated by a plus or minus number in the upper right corner. You can correct it forward or back by several days if you so choose, but you can only move it one day during each display. If you do not want to make any changes or have already entered a correction and do not want to change it, just hit ENTER.

In the upper left corner, you will be asked if you want to see the next month. Press anything but Y plus the ENTER key and the computer will take it as a no answer. If you choose Y, the computer will advance to the next month and present its critical days. You can continue the process indefinitely, but you do have to know if a year is a leap year as you will be asked that every February.

If you choose not to continue, the program asks if anyone else wishes to run it. Choosing Y here sends you back to the start, allowing someone else to input their dates, or allowing you to switch over to the other mode of presentation, in this case, the graphics mode.

If you now press G, you are again asked for the five dates. The computer will draw a solid white line in the center of the screen. This is the critical day line. Along the bottom is a chart of the days of that month. Above the white line is positive and below is minus.

The lines for each cycle will then be drawn. The red line is the physical cycle, the blue is the emotional, and the yellow is the intellectual. Notice that when two lines intersect or a line crosses the white line, it tends to spread out a bit. This is what I referred to earlier. It is also why I did not bother to add any correction factor to the graph.

Once the lines are drawn, you can easily advance to the next month, or go back to the beginning, or just end the program. It is not a difficult program to run.

Line 10 of the program sets several variables to zero, which allows you to change presentations without having to end the program. Line 15 asks for choice of presentation and 20 is an error trap. I normally put in error traps wherever possible, but since I was trying to keep this program under 2 KB, I could not do that everywhere. For the same reason, I did not use "prettyping" and did combine many statements.

Line 25 has you input the dates needed to figure the cycles. The computer must calculate how many days you have lived. There are three possibilities here: your birth month is either the same as or comes before or comes after the chart month (i.e., if your birth month is May, the chart month would be either May, July-December, or January-April). If both months are the same, line 30 jumps over most of the calculations. If your birth month is before the chart month, line 35 sets some variables and advances to the proper section. Otherwise, lines 40-50 add up the number of days in each month to the end of the past year, starting at and including your birth month. Line 45 sends you to a list containing the number of days of each month, which becomes the G variable. Lines 55-75 add up all the days of the months that have passed in the chart year, not including the chart month. This is the L variable. Note line 60, which comes into play if the chart month is January and must be handled a little differently.

Line 80 subtracts the unlived portion of your birth month (i.e. if you were born on the 10th, the dates from the first to the ninth would be the unlived portion), and makes that the H variable. It also sets F equal to the number of years lived. Note in line 85 that if your birth month comes after the chart month, F must be decreased by one. For example, if you were born in December 1952 and want to figure out your cycles for September 1983, you would subtract 52 from 83 to get your age, which would come out as 31. But you will not turn 31 until December, so your age is actually 30 until then. This is what line 85 corrects.

In line 90, the number of years lived is multiplied by 365.25, the number of days in a year, and the G and L variables are added in. The .25 is intended to account for leap years and will add an extra day for every four years. But because it increases by every four years and not by counting actual leap years, the total may be off by a day. The correction factor can take care of this, if necessary.

Line 95 figures out the state of each cycle at the beginning of the chart month. This is accomplished by dividing the total days lived (F) by the number or days in each cycle. The answer is then reduced to a whole number by the INT command, and that number is multiplied by the number of days in the cycle. The resulting number is subtracted from the total days lived, and what is left is the number of days into the current cycle at the beginning of the chart month. If the number is five, you are in the fifth day of whatever cycle is being calculated. You are also in the plus or positive phase since you have not hit any of the halfway or changeover marks. The variables for the physical, emotional, and intellectual cycles are M, N, and O, respectively.

Having obtained a starting point, the cycles must be extended through the rest of the month.

While the section is long, the process is simple. For each calendar day that passes, the computer adds one to the cycle. The computer will constantly check to see if it has reached either the halfway points (11.5, 14, or 16.5 days) or the end points (23, 28, or 33 days). If it has and you have chosen numbers, that date is displayed under the appropriate heading along with a plus or minus sign. These are the critical days, and the signs show what phase the cycle is in. This is done until the day of the chart month is reached.

Line 100 only comes into play when you want to advance the program beyond December. It adds one to the year and resets the program to January of the next year.

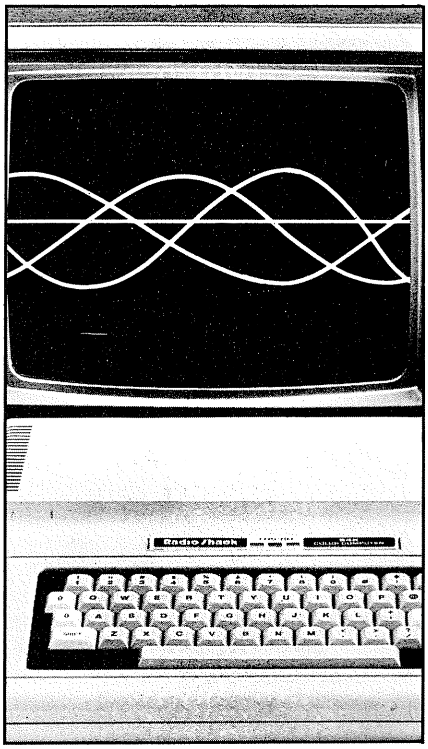
Line 105 sets up the display if the chart or numbers have been chosen. Otherwise, line 106 sends you to the graphics section to set up the graph. The R variable is used to actually position the number and signs on the screen. Line 115 determines how many days are in the chart month, and 120 will add a day to February if you reply that it is a leap year. Line 125 again checks to see if you have chosen graphics, and if so, sends you to the proper section.

If you have picked the chart, line 130 sets up a loop for the total number of days for whatever month is being calculated. Lines 135-145 check to see if any cycle has run its complete course, and if so, sets it back to zero. Line 150 then advances each cycle by a day. If you are using graphics, line 155 will send you to the proper routine and then send you back to line 130 to check out the next day.

If you are using numbers, lines 160-165 determine if the physical cycle is in either the plus or minus phase and if it has reached either the midpoint or the end date. If it has hit either of those dates, that date and the sign are printed. (Note the use of the R variable to position the sign above the date and in the correct column.) Lines 170-175 do the same for the emotional cycle, and 180-185 handles the intellectual cycle. Line 190 completes the loop and sends it back to move onto the next day, which is the P variable.

To keep things simple, I rounded off the physical and intellectual cycles' halfway points, from 11.5 and 16.5 to 11 and 16, respectively. Any error this introduces is small and if it bothers you, use the correction option to change it. Also note that with this setup, the information is displayed all over the screen with the earliest critical day being the highest on the screen and each following critical day a little lower down, regardless of what cycle it is in.

Lines 195-210 are the correction factor routine that allows you to move the cycles forward or back. If you do enter a correction, the program will check to see if it is forward or back in lines 200-202. The program then jumps to line 210 to add a day to or subtract one from each cycle, and then will automatically advance the program to the next month without further input from you. Once a correction is entered, line 215 ensures that it stays in until you



decide to remove or modify it. Line 215 also comes into play if you choose not to make a correction.

Line 215 lets you advance to the next month. If you choose not to, then line 220 lets you restart the program. This lets you change over to the other form of presentation or lets someone else check their biorhythms. Line 225 ends the program and prevents it from running into some subroutines.

These subroutines start with line 230, which sets up the chart's display heading. Lines 235-245 provide the number of days for each month of the year; it's used a number of times in the program.

Line 250 starts the graphic display and draws the white zero line. Line 255 displays the chart month and year at the top of the screen and positions the plus and minus signs. Lines 260-275 print the calendar day along the bottom of the screen, showing as many as needed for each month. The only exception is February, which will always list as 29 days long along the bottom, but the cycles will not reach the last day unless it is a leap year.

Lines 285-300 actually draw the cycles on the screen. Let's examine lines 285-288 in detail as these lines control the physical cycle. The key to this is to break the sine wave into four quarters. The first quarter is in the plus phase and will go up until it reaches the second quarter and heads down. The line continues going down even after hitting the third quarter, which is when it crosses the zero line and becomes negative. The line does not start upward until it reaches the start of the fourth quarter, but it is still in the minus section. When it reaches the zero line, it has completed a cycle and is now back at the first quarter.

If you understand that, you will see that what I had to do was break the sine wave into the four quarters and find some formula that would calculate the y component for each quarter. That is what lines 285-288 are doing for the physical cycle. Lines 290-293 handle the emotional cycle, and lines 295-298 take care of the intellectual.

Look at line 285. This says that if the physical cycle (M) is in the first six days of the cycle, then the y component (AM) will be a negative number. Now jump to line 300. This line actually sets the block for all three cycles. Note that AM is added to a base of 15, which is the middle of the screen, so if it is negative, the line will go up. To make the line descend, AM must be positive, which is taken care of in lines 286-287. Line 288 makes it positive again for the final quarter.

As I mentioned earlier, line 300 sets the points. Note that to move across the screen one day, you must move two points. This is caused by the PRINT@ statements taking up two points for every one date. Line 305 returns you back to the main loop.

That just about covers the program. Regarding improvement, the obvious thing would be to rig up a printer output, but I must leave that to someone with a printer. As for the MC-10, all commands found in the program are in the MC's Basic. I made a deliberate effort to keep all lines less than 128 characters long, which is the MC's limit. As far as I can determine, the PRINT@ and the graphics screens are the same, so I think the program will run without change on the MC-10.

Robert Toscani has published numerous computer program articles. Being a private pilot, he has seen more than 100 of his aviation articles in print.

(Program listing on page 136.)

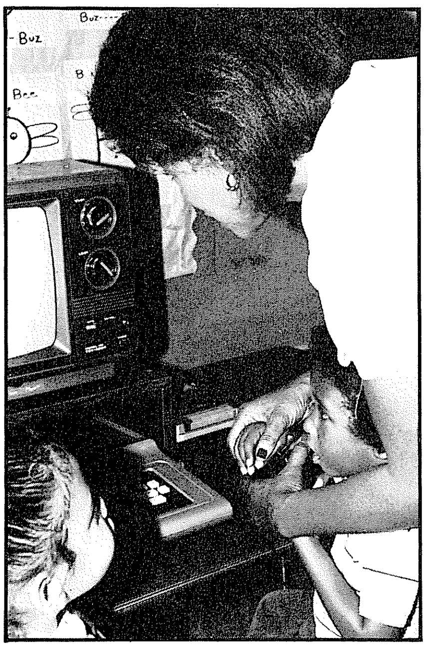
Computers: Power Tools for Education

The greatest advantage for teachers and students is the computers' inherent capacity for non-critical, untiring repetition.

by McAlister J. Merchant, Jr.

There are many programs available using varying approaches to computer-assisted instruction (CAI). These approaches include page-turning techniques, various types of simulation, and trial-and-practice. Many teachers are justifiably derisive of the use of computers as an element in page-turning techniques; after all, flash cards and workbooks *are* much less expensive. Each of these various approaches to CAI does, however, have its place and enhances the learning environment when properly used. Children learn best through experience, and the correct computer hardware/software system can add a new dimension to that experience.

Recent experiments with preschool-age children demonstrate some advantages to using computers with this particular age group. The greatest advantage for the teacher *and* the student is the computers' inherent capacity for non-critical, untiring repetition. These children are pre-literate, that is, they are generally capable, though limited, verbal communicators. Their command of the tools of literary communication—their symbological vocabulary—though growing daily, is severely limited. The process used to change this situation involves exposure and practice, enhanced by repetition and a positive physical environment. Children at this stage of development are developing rapidly both physically and mentally and the two areas are intimately related.



As any parent of a two- or three-year-old knows, the child's search for knowledge—the what and why of everything—is relentless. Children really do want and need to learn. They love attention and they seek control.

The control aspect is a very important part of the learning process. What children learn from us directly, indirectly, purposely, or inadvertently regarding means toward and degrees of personal control are some of the most effective aspects of this early learning process. If a small child learns that adult attention is gained by throwing toys across the room, it will throw toys across a room tirelessly for what may seem like hours. A child receiving elaborate praise for pointing to a funny-looking spotch on a piece of paper and saying "two," will appropriately (or inappropriately) point and say "two" just as long as the adult pays attention. All of this attention-seeking and attention-rendering is time consuming, and one of the painful realities of modern life is that we cannot always give our young children all the direct attention we might choose to. There is a recent television commercial that shows a charming preschooler happily learning to use a computer in the arms of his father. Though this is an endearing image, experience demonstrates that life is not that simple.

We, as adults, too seldom have the time and very rarely have the patience for this kind of teaching. With careful

planning and some loving input from us, however, the computer can become a surrogate, welcomed by us and by our children.

When a child presses a button on a computer, something happens. Here is a televison picture that he or she can change instead of just watch! With the proper program, when a button is pressed, colors, pictures, and sound go wild all over the place—it's really exciting. And the computer never gets tired.

There are far too few programs for young children (ages 2-6), but if you have a Color Computer you can easily write your own.

There are, besides the usual good programming techniques, some special considerations for computer programs that can be successfully used by preschoolers. These same considerations make ideal parameters for evaluation of commercial programs.

Programs should focus on a single simple concept. Possible subjects might include any one of the following: primary colors, numbers, names of household animals, kitchen appliances, things that are green, small things, big things, etc.

If the concept is broad, it should be handled in sections, and maintain specific narrow ranges. Recognition and response type programming is best for those children who do not yet read. A program that displays an object or part of an object, in graphic form and then requires a simple "go"/"no go" response is best. In other words, though hunt-and-peck typing may serve you or me for necessary input routines, it may only frustrate a small child. Perhaps the choice between "the big key" (space bar) and "any other key" would be better. This is especially true with smaller children whose motor skills are not yet finely tuned. Children with more knowledge and training can more readily distinguish colors, numbers from letters, and specific characters from each other. It is also true that matching screen characters to keyboard characters is better practice than unprompted choices.

The child should have a great amount of program control. Involvement is the name of the game.

Limit the number of operational keys for any given program. Use key groups rather than the full keyboard. Use fewer keys for younger children. If you're working on numbers or only what number-key responses, design the program to ignore responses other than numbers without crashing the program.

Use large simple characters. Fancy letters and numbers may seem to dress up a display, but, for the most part, they confuse the early learner. Roman letters and block numerals are best for

Variable	Description
P	column print location
R	row print location
I	general loop counter
T	SET/RESET code index (pulled from data list)
S	sound pitch
Q	timing loop (pause) counter
L	previous number selection
N	current number selection
LL	length key for LL$
CC	color code
A$	general input string
ND$	marker character, *
LL$	marker string
N$	input string for number choice (converts to N)

Figure 1. Variable List

control selection, and recognition-matching programs are best for the youngest children.

Use color, animation, and sound. The success of arcade games and Sesame Street have much in common. Movement, pace, color, consistency of result, and sound are all important aspects of both. These are the rewards of play—the payoff. Use them often, separately and together. Don't reserve them until the end of the game. Use little payoffs along the way. Remember, the goal seems immensely more desirable if the trip itself is rewarding.

Use the joystick purposely. The joysticks are best used when one of the goals of the program is development of eye/hand coordination. They also are helpful in improving writing and drawing skills.

The TRS-80 Color Computer is well-designed for use as a tool in early childhood education because of its powerful graphic capabilities and the ease with which it can be programmed for such use. The accompanying program illustrates the principles I've outlined above, with one addition—verbal communication.

The program is a tool for teaching numbers and some techniques of computer usage. The child is rewarded after each selection by a quasi-random visual and audio display. The program is written in extended Basic because I like the graphics capabilities, but it is easily adaptable to the SET/RESET graphics of non-extended Basic. I've also added a little trick that makes the computer "talk" to the youngster, telling him or her what number was selected at the same time it is shown on the screen. This feature will also work in non-extended Basic. This routine allows us to bridge the literacy barrier and also permits the child to search and try without our intervention—an exercise and expansion of personal control.

Program Outline

5-40	Introduction and loading instructions
50-100	Opening title
100-130	Title sounds
135-175	Data list for opening title
190	Pause for reading title
200	Reset data
210-230	Clear screens
240-250	Opening instructions
255	Set previous number variable
260-290	"Numbers used" marker set
300	Number selection & audible prompt
310-330	Control for accessible numbers (320) and functioning keys (330)
340	"Numbers used" marker string
350-370	"Selection made" response
380-390	Locate endless loop position of correct recording
395	High speed ON
400-430	The computer speaks
435	High speed OFF
440	Move through data list to position
460-500	Set screens, draw number, and pause
505	High speed ON
510	Change background color
515	High speed OFF
520-530	Change all colors then pause
540-620	Data list for large high-resolution numbers
630	Final display audio
640	Reset sound prompt variable
650	Reset data list
660	Number input control upper limit
670	"Numbers used" marker locator
680	Ending audio sequence
690	Do it again

The talking computer section requires an endless loop cassette tape like those used in answering machines. A 60-second tape is good. A longer tape will make the waits between outputs rather long; a shorter one may not allow enough space for all the numbers.

1. Prepare the tape by inserting the endless loop cassette into the recorder and locating the start. You should determine the length of the tape in counter digits on your machine so that you won't overwrite the beginning. This can be done by OPENing and CLOSEing a null file, then typing AUDIOON:MOTORON. Run the tape forward through the

OPEN/CLOSE cue twice, noting the counter digits (on my RS CTR 80A, a 60-second cassette is 94 units long). Remember that rewinding the endless loop tape may destroy it. After you have determined the tape length, erase the tape and then listen for the start point of the tape and stop the motor.

2. Set a marker for each recording by OPENing and CLOSEing a file named for each number. Example:
OPEN "0", - 1, "1":CLOSE-1 (ENTER)

3. Disconnect the AUX and MIC jacks from the recorder and *immediately* announce the number. Example:
"Your number is one"

4a. Immediately reconnect the two jacks. *Note:* Disconnecting the MIC jack starts the tape and reconnecting it stops the tape. The more skillfully you perform this sequence, the less tape you will use.

5a. Use MOTORON to advance the tape about one tape-counter digit, then hit any key and ENTER to stop the motor.

An alternate method that will create a shorter tape is to perform all the motor advances *except* OPEN/CLOSE file statements manually from the recorder. Follow steps 1-3 above, then 4b and 5b below. Reduction is about 10 percent.

4b. Immediately reconnect the MIC jack to stop the motor.

5b. Use the fast-forward key of the recorder to advance the tape one counter digit.

6. Repeat steps 1-5 for all numbers.

Shorter statements permit more inputs or use of shorter length cassettes. Consider the following:

"Your number is one"	(5 syllables)
"You've chosen one"	(4 syllables)
"You chose one"	(3 syllables)

Use sentences, but short sentences.

Recording the numbers in ascending order will give the quickest response if the numbers are chosen in counting order. Rearranging the sequence can optimize response times for more widely separated numbers. Try the sequence 2,7,4,1,8,5,2,9,6,3.

McAlister Merchant is a consultant product designer specializing in the design of housing for computers and other electronic equipment. He has written several essays on computers as a design tool. He has also submitted a proposal to the National Endowment for the Arts on the same subject.

(Program listing on page 143.)

Computers in Early Childhood Education

Looking through the phone directory for I can't remember what, I ran across an ad for a daycare center and nursery school that included computer education as part of its early childhood development program. I was flabbergasted—computers for preschoolers?

In Ted Nelson's *Computer Lib/Dream Machines*, he described the computer as a toy of children in this, the best of all worlds to come. But after several years of awaiting Xerox's Smalltalk and Alan Kay's Dynabook, the dream seemed to have dimmed. Was it possible, I wondered, that someone was finally making the dream come true. You better believe it.

I've long since ceased to be amazed by the capabilities of children, but I'm always surprised when I find other adults who believe that kids are people too—adults who act on the understanding that these are vital minds that are always questing, digesting, and testing.

Two years ago when Debbie and Floyd Stevens started their school, their guiding thought was to create an ever-expanding environment of discovery and experience. They wanted to provide a truly enriched learning experience.

Almost everybody loves a good arcade game. Floyd looked around one day and realized that there was a vast amount of untapped power in the skills required to win at the better video games. Here were children practicing analytical and deductive skills for which many of them found no outlet outside of the arcade.

In pondering the question of how to better prepare these young people for successful adult lives, Floyd realized that the place to begin is at the beginning. The understandings we gather as young children shape our lives forever after. It was at this point that he and Debbie decided to institute a computer program at their California Center for Early Learning in Los Angeles, CA.

Last year, Floyd began the investigation and evaluation process necessary for proper implementation of any computer education program. Since he already had a good idea of what they wanted to do with the system, his next step was to check out different computers.

"We, of course, looked at Apple. You can't avoid it. The first thing we found out was we couldn't afford Apples for our students in the type of program we wanted.

"Next, we looked at Atari, the 400 and the 800. The 800 was OK, but I was concerned about the hardware

restrictions. The VIC 20 was a toy and the Commodore 64 didn't have enough programs or service centers.

"Radio Shack seemed to have the right equipment, especially when I was told about the impending memory capacity upgrade for their Color Computer. But I think the thing that did most to sell a TRS system to us was the attention we got and continue to get from all levels of Radio Shack people."

Through conversation with Terry Kramer, the West Coast Regional director of Educational Marketing, and their local Computer Center manager, the Stevens received advice on system requirements and software inventory.

Floyd says that even though all their equipment is installed, paid for, and no longer under warranty, his "hotline" to Tandy is still well tended at the other end. He says that he gets good service whether it's from his nearby Computer Center or from Radio Shack's Dallas, TX headquarters.

There are few guidelines or evaluation standards for early childhood computer-assisted education. In fact, until fairly recently, there were few programs available for young children. So Debbie and Floyd spend quite a bit of time sitting at CoCos, personally evaluating programs relative to their overall program guidelines.

Their aim is to bring into the classroom as much experiential learning as possible. To that end, the school has two 32 KB extended Basic CoCos with tape recorders, one 64 KB Model 4 with two disk drives, and a DMP 200 printer.

The programs they use are oriented toward teaching colors, numbers, shapes, and Logo. The best programs for the lower age groups are more heavily weighted toward the pictorial than to word text—a situation for which the CoCo is well suited. Though the Center for Early Learning is primarily a preschool daycare and early childhood education center, it also has programs for school age children who participate after school hours and on Saturdays in computer orientation and programming classes as well as classes in experimental electronics. Tutoring in more traditional subjects is available, but the demand for it is nowhere near the demand for the computer classes.

Besides the obvious results of the program, including the understanding of computer operation, the Stevens have discovered that there are some very valuable side benefits. The attention span of the younger children who have been using the computers for any length of time is much greater than that of children of the same age who have not been working with the computers. The self-control of youngsters who are doing at least part of their learning through the use of computer programs is greatly improved in a short period of time. Debbie believes that this is due to the amount of control the children can exercise over the computers.

With such impressive results, it's easy to see that there's a bright future for computers in early education programs—you can see it reflected in the eyes of the eager youngsters at the Center for Early Learning. —MM

How to Define Computer Data

by Edward L. Cound

We continue our look at the fundamentals of programming in Basic, this time by examining the different types of information that a computer can deal with and how it does so.

A	B
Numeric	Constant
String	Variable

Whatever type of information you wish to have a program deal with, it helps to realize that it can always be classified as one item from column A *and* one item from column B. Thus, only four combinations are needed to define all the different types of data computers can deal with: numeric constants, numeric variables, string constants, and string variables.

Let's take a look at examples of each. Type this into your computer:

PRINT 3

What do you get after you press ENTER? Hopefully, just the number 3. This is a numeric constant. It is numeric—that much is obvious. And it is constant—that is, 3 is always going to be 3. If I have three apples in front of me, I cannot make those three apples into four apples or two, or anything else, without adding or subtracting apples from the original three. Constants are pretty easy to deal with.

Only four combinations are needed to define all the different types of data computers can deal with.

But if the computer dealt only with constants, it would be little more than a glorified (and expensive) calculator. Most calculators, after all, deal only with constants; as you press numbers on the calculator, you are actually keying in the constants that you want to compute.

But part of the power of the computer is its ability to deal with variables as well as constants. Try this:

PRINT A

This should put a 0 on your screen. Now try typing:

A = 3

And then enter:

PRINT A

Now, a 3 should be printed out. You have assigned a value, 3, to a variable, A. Try this:

A = 7
PRINT A

You have changed the value of variable A—it has *varied* from 0 to 3 to 7.

What is a variable? Variables are just receptacles for information. A variable is a name to which we assign a piece of information. The information can vary over time, just as the value given to the name A above varied.

When we give the computer some information to remember, it goes into a section of memory. Where, you wonder? Fortunately, that is up to the computer.

It helps to think of computer memory as nothing but a bunch of boxes, each box containing one character (one byte). The boxes are arranged in one long row, numbered sequentially from 0.

In most of the computers that we are talking about, there may be anywhere from 16,000 to 128,000 of these memory "boxes"; this is the total of the machine's RAM and ROM memory.

Assume that you want to store a number, say 3. The computer might place it in box 31208. Now, what if you had 10 numbers to store? They might be stored in boxes 31208, 31209, 31210, and so on. You can see that as you increase the amount of information you wish to deal with, the process of remembering all those box numbers, or *addresses*, becomes problematic, to say the least.

What if, in addition to everything else you have to keep track of when programming, you also had to remember the address of each piece of information stored inside the memory. In the early days of computing, before high-level languages like Basic were developed, that is precisely what was done.

Information was put in specified locations, and you had to remember where it was.

Things have improved substantially. Rather than remembering that the number 3 was stored at 31208, we now have the computer store it wherever it likes, and tag it with a name. The computer looks it up under that name, resolves where the address of that information is, and retrieves it from there.

So, when we say A = 3, we are telling the computer to take this number (3), put it somewhere in memory, remember where it is, put a name tag (A) on it, and remember that this name tag refers to that address. We don't have to deal with all of the overhead entailed in remembering the addresses of all of that information; the computer remembers for us.

In order for us to store a particular piece of information inside our computer, we have only to devise a variable name to represent that information. Then we assign our information to that name, using one of Basic's assignment statements.

How do you choose a variable name? It used to be quite simple. In early Basic interpreters, you simply picked a letter, A-Z. There were only 26 potential variable names, and you tried, if possible, to pick one that meant something, one that helped you to remember what it was for. When tackling a problem involving an amount, an interest rate, and a period, you might choose A, I, and P. If you also had a payment, you couldn't use P for both period and payment, so period might be called Q or something else.

The situation has become more complicated. Most TRS-80s will allow, at the very least, two-letter combinations, or one letter followed by a number. This yields A-Z, AA-ZZ (AA, BB,AC. . .AZ,BA,BB . . .etc.), as well as A0-A9. . .Z0-Z9.

In other words, you are allowed variable names with two significant characters. You may use more, but only the first two count. AARDVARK might as well be called AA, unless you want the additional description. And RADIO, RADIUM, RADII, and RADIAL are all considered the same, because their two significant characters are RA.

This method gives you hundreds of variable names from which to choose.

Still, in many advanced versions of Basic, including the new Model 4 Basic, much longer names are allowed, up to 40 significant characters in Model 4. Given our example above, we could now have variables called AMOUNT, INTEREST, and PERIOD. The program becomes a bit more verbose and it requires more memory to store those names, but it is much easier to read the program and its purpose may be more clear.

If you are not sure what your machine will allow, check your Basic reference manual to find out what constitutes a legitimate variable name.

Note that some letter combinations are not allowed; words that Basic understands as part of its syntax, like PRINT or REM or INPUT, are called Reserved Words. You cannot use these as variable names since the computer understands them as something else. A complete list of reserved words will also be in your reference manual, but this isn't something to worry too much about at this point.

So, when embarking on a programming task, choose your variable names appropriately. If you are using shorter names, like A, I, P, and Q, define them completely for future reference, either in a notebook, or using the REM statement. REM, short for REMARK, allows you to leave comments in

the test of your program. For example:

10 REM   I is interest, A is amount

This will REMind you later on of what these short variable names represent.

When your information is non-numeric, it is referred to as string information. Let's say that you wish to store the name John Doe in memory. Using our analogy of memory as a series of boxes, 1 byte (or character) per box, we see that the first box used will be for the J, the second box for the O, the third for H, and so on, until we get the E into the eighth box (yes, the space betwee JOHN and DOE is a character, too). What we end up with is a string of boxes—a J strung to an O strung to an H strung to an N, etc. This is why non-numeric data is called string information.

Try this:

PRINT "HELLO"

Whatever you typed between the quotes should be repeated back to you on the screen. If you typed "HILLOOO" by mistake, that is what you got back. The computer will print literally what you put between quotes (which is why strings are sometimes also called *literals*).

"HELLO" is a string constant. Like the numeric constant, this is a value that remains constant. String constants are always contained inside quotation marks.

But we can use string variables as well. Try this:

```
A$ = "HELLO"
PRINT A$
```

Dealing with strings may even be easier than dealing with numbers, but you do deal with them a little differently.

First, we need to differentiate between variables that are used to represent numbers, and those used to represent strings. You choose a variable name, as described above, then you add the dollar sign ($) to it. Voila—you have a string variable name, as in A$ or YR$ or NAME$. The $ tells the computer what type (string) of information will be assigned to the variable name.

And that's all there is to it—one from column A, one from column B.

Let's finish off with a program that uses several variables, as well as Basic math operators (+ , –, *, /).

```
10  CLS
20  INPUT "WHAT IS YOUR NAME"; N$
30  PRINT "WHAT YEAR WERE YOU BORN IN"; N$;
50  Y = 1984-Y
60  CLS
70  PRINT N$; "YOU WILL BE"; Y; "IN 1984"
80  M = Y * 12
90  D = Y * 365
100 PRINT "THAT MAKES"; N$; "ABOUT"; M; "MONTHS
    OLD"
110 PRINT "OR ABOUT"; D; "DAYS OLD!"
```

This should give you something to gnaw at for awhile; it's simple, but effective. Again, as I've said before, experiment with it, toy with it. Make it compute minutes and seconds as well, or compute for two people at once. Try changing the printouts. Whatever you choose will give you something to work at, building a foundation for your programming skills.

Get to it—that is how you will learn it.

Ed Cound is a graduate in English/Communications at the University of Minnesota. In addition to his technical writing, he is a consultant and programmer specializing in interpreter and compiler Basic, and Z80 Assembly language.

The Dynax DX-15: An Affordable Daisy Wheel Printer

Although it's rather slow, the DX-15 can provide high-quality printouts at a bargain price.

Reviewed by Richard Green

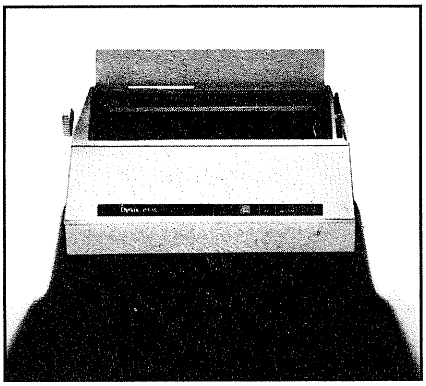
The Dynax DX-15 continues the current trend of adapting typewriter components to make computer printers. Like Smith-Corona's popular TP-1, the Dynax DX-15 (made in Japan by Brother) is essentially an electronic office typewriter that has been stripped of its keyboard and has had an interface added.

The DX-15 retains most of its typewriter qualities. The ribbon lift includes a reel for correction tape that may alternately be used for red-colored ribbon. Because of this feature, the DX-15 can print in two colors. The main ribbon is a standard Brother cassette and is available in several colors at most office supply stores.

I was rather bemused by the existence of a correction ribbon on a printer until I discovered that the DX-15 has a separate keyboard available as an option. With the keyboard connected, the DX-15 can be used as a typewriter.

Also available as options are a sheet feeder and a tractor attachment. Either simply snaps onto the printer, but neither is intended to be installed by the user. Initial installation must be done by a dealer. Once installed, the sheet feeder or the tractor may be removed to allow the use of single sheets.

The DX-15 can be used with any of the popular microcomputers because it is available with either a parallel or a serial interface. The parallel interface is Centronics-compatible. It was able to plug the printer into both a TRS-80 Model III and an NEC Advanced Personal Computer without any modifications to the printer cable. Of course, the instruction manual does specify the wir-



ing of the cable, so it should be a simple matter for the dealer to make up a cable for any specific computer. The serial interface is RS-232-compatible. A set of switches on the back of the printer allows all the parameters of serial transmission to be set without disassembling the computer. The DX-15 can be set to accept data at any of eight speeds, ranging from approximately 140 words per minute (110 baud) to approximately 12,000 words per minute (9600 baud).

The speed at which the printer accepts data has no effect upon the speed at which it prints. Even the slowest data rate is nearly twice the printing speed. With an appropriate program, however, the higher baud rates could be advantageous. As a background task, the buffer of the DX-15 could be periodically filled by the computer, allowing it to be used for other jobs while printing continues.

Like most of the printers being sold today, the DX-15 is microprocessor controlled. It has a 3 KB buffer that may be increased to 5 KB. Multiple copies can be made of any document that will fit into the buffer.

Line spacing may be set to single, 1-1/2, or double space by means of a touch switch on the front panel of the printer. Any other line spacing, in increments of 1/48 of an inch, can be set through software commands. The paper can be adjusted manually by pressing in on the platten knob while rotating it.

The DX-15 can produce four types sizes: pica (10 characters per inch), elite (12 characters per inch), report (15 characters per inch), and proportional. Selection of type size is made by pressing a touch switch on the front panel. As the switch is pressed, a light illuminates indicating which pitch has been selected. Of course, the print wheel must match the pitch selected.

There are 116 different print wheels available for the DX-15, including one that is an electric collection of symbols and Greek letters. Special purpose print wheels include ASCII, English, French, Spanish, Norwegian, Italian, Swedish, Finnish, and Portuguese. Additionally, there are specific print wheels for various dialects of these languages, such as American Spanish and Swiss German. The print wheels are enclosed in a clear plastic cassette that is simply inserted into the print head assembly—it's not necessary to touch the print wheel to change it, so you won't get inky fingers.

Fifty-two software commands are used by the DX-15. In addition to standard commands such as line feed and carriage return, the available commands allow reverse line feeds, back spacing, vertical and horizontal tabs, and switch-

ing among the four type sizes. One interesting feature among these commands is called "shadow print." When this is selected, any time a character is double printed the carriage moves 1/120th of an inch to the right before making the over strike. The result is a very dark bold-face print. There are also a half line feed and a half-reverse line feed that allow both subscripts and superscripts to be printed.

Automatic underlining is also included. When this is selected, all characters, except blanks, are underlined as they are printed.

Margins may be preset on the printer by positioning the print head where the margin is wanted, and then sending a control code to the printer. Separate codes are used for right and left margins. Once set, the margins remain until the printer is either turned off or a new margin is set. Top and bottom margins are set similarly. The paper must be positioned where the margin is desired and an appropriate code sent. As with the left and right margins, the top and bottom margins remain until they are deliberately changed or until the printer is turned off. The top and bottom margins may be automatically set to 1" by the positioning of a switch on the back panel of the printer. This is primarily intended for use with fan-fold paper to skip the perforations between sheets.

The one great disadvantage of the DX-15 is its speed. It is s-l-o-w! The instruction manual claims a maximum speed of 13 characters per second, or about 130 words per minute. The printer did not reach this speed in any of the tests that I ran. A more realistic number would be around 10 characters per second. I found this surprising, considering that it prints bi-directionally and is supposed to have logic to seek the shortest printing path.

Use of the function switches on the front panel of the DX-15 is very confusing. There is no CLEAR button at all. To clear the buffer, the instruction manual directs you to hold down the COPY switch, then press the SEL switch. Although clearing this way isn't readily apparent, you could become accustomed to it, except for two things. First, it doesn't work. Second, if you reserve the order in which you press the keys, and hold the SEL switch, then press the "COPY" switch, the printer quits printing and quits accepting data from the computer. Pressing the SEL switch a second time allows the printer once again to accept data from the computer. Printing continues from where SEL was pressed. The buffer is *not* cleared! In fact, there is no easy way to clear the buffer other than to remove power from the printer. But if you clear the buffer in this way, you also clear every parameter that you've set: margins, tabs, line spacing, and print pitch!

Surprisingly, the DX-15 does not have a reverse form feed. If you want to print in columns, it is necessary to manually align the paper for the second column. I attempted to print two columns using the reverse line feeds. The results were unacceptable.

> The DX-15 also has proportional spacing, which is a rarity on a low-cost daisy wheel printer. With a good word processing program to furnish right-margin justification, the DX-15 can produce print that appears to have been typeset.

The paper creeped on the platten enough that the print of the two columns was obviously out of line.

During the time that I had to use the DX-15, I found the internal buffer to be a constant irritation. It is not big enough to hold a complete manuscript for an article, so the computer was tied up with printing just as long as it would have been without the buffer. Also, because the only way to reliably clear the buffer is to turn off the printer, each time the buffer is cleared, all print parameters must be re-entered.

The literature that accompanies the DX-15 is not much help. The instruction manual is a thin little 46-page booklet that is accompanied by an even thinner booklet of changes to the original manual! Considering the small size of the instruction manual, it is incredible that it wasn't simply reprinted.

Although the DX-15 has many "smart" functions, the manual does not include any attempt at a tutorial. If you have used advanced printers (i.e., an Epson), you'll have no trouble deciphering the instructions. If the DX-15 is your first printer, however, I'd strongly recommend that you borrow someone's Epson manual. Although the codes for the Epson and the Dynax are not the same, they are implemented identically. The Epson manual contains an excellent tutorial that will illustrate the techniques needed to use all the features of the DX-15.

The greatest advantage of the Dynax DX-15 is its low cost. The suggested retail price of the DX-15 is only $599 with a parallel interface. I have already seen the printer advertised at $475. This would be cheap for a quality dot matrix printer, but it is nearly unbelievable for a daisy wheel printer.

The DX-15 also has proportional spacing, which is a rarity on a low-cost daisy wheel printer. Truly impressive results can be achieved using this option. With a good word processing program to furnish right margin justification, the DX-15 can produce print that appears to have been typeset.

The cartridge daisy wheels are a positive advantage. The changing of the wheels is the simplest of any printer that I've used. The print wheel cartridge is simply slipped into position. When the carriage is closed, the wheel is automatically driven to the correct position. There is nothing to align. According to the manual, each daisy wheel is good for 10 million characters. Obviously, I didn't test this claim; at the speed the DX-15 runs, it would take about 12 days of non-stop printing.

The Dynax DX-15 is the lowest-cost daisy wheel printer currently available. At the prices now being advertised, it is at least $150 less expensive than any other letter quality printer available. And the quality of its print is as good as that of any other printer.

It is slow, though. At a tested speed of approximately 10 characters per second, it is the slowest daisy wheel printer that I've used.

If you have a need for a small amount of correspondence or program listings, and speed is not especially important, the DX-15 should be considered. It offers the greatest number of features at a bargain price.

Product:
Dynax DX-15

Manufacturer:
Dynax
5698 Bandini Blvd.
Bell, CA 90201

System Requirements:
Centronics Parallel or RS-232 Serial Port

Special Requirements: None

Price:
$599 (parallel), $649 (serial)

Quality Printing at a Reasonable Price

Whether for business or home use, the Bytewriter is a quality printer/typewriter that can be connected to a large number of different computers.

by Patricia Steele

A few years ago, I had to make a choice when I purchased my printer. Did I want speed or did I want a letter quality printer? Everytime I assembled a program or listed a file, I was convinced that I had made the right choice when I decided upon the TRS-80 Line Printer V. I never had an article rejected because the manuscript was printed with a line printer, but after I packaged the material to send to an editor, my thoughts would return to daisy wheel printers.

I was also spending more time revising a makeshift word processor than I spent on writing articles. I decided that it was time to concentrate my earnings on tools to work with.

I had the new release of Pickels & Trout CP/M, 2.2eH – Cameo, and was informed that Sorcim had a word processor called SuperWriter that would run under this version. I ordered the software and spent the next few days going through back issues of computer magazines. I read everything I could find on daisy wheel printers and stopped when I came across an advertisement for the Bytewriter. The ad indicated that Williams Laboratories had added a microprocessor-controlled interface to the Olivetti Praxis 30 portable typewriter and turned it into a letter quality printer. The ad also stated that the Bytewriter functioned as a typewriter with or without disconnecting it from the computer. I wanted to find out more.

Not only did Williams Laboratories respond with an immediate reply to my inquiry, they responded with a pleasant surprise. If I ordered the Bytewriter directly, there would be a $12 shipping and handling charge, but they offered the Bytewriter at $495—a $200 savings from the ad price.

The company also carries the interconnecting cables for my TRS-80 Model II ($39). I purchased a money order, not wanting to wait two weeks for my check to clear, and a week later I had my Bytewriter.

Fortunately, the Bytewriter was shipped with a film cartridge ribbon. If it had been shipped with a nylon ribbon, I may not have found out that there was something mechanically wrong with the typewriter until one day past the warranty. The guide behind the cartridge housing was slightly bent. This caused carbon to be scraped away in several places on the film ribbon, resulting in poor print.

I had the experience of being another dealer's customer when I had my car serviced at a dealership other than where I purchased it. My priority in the queue was altered so many times that I had to return the next day and try again. It was just the opposite at Schauers Office Services near my home. They did indeed honor the 90-day warranty on the Olivetti and were very courteous and helpful. They answered all my questions and demonstrated the right way to insert the ribbon cartridges and print wheels.

All of the Bytewriter's hardware is internal and removable by unplugging. There is a 90-day warranty on the Bytewriter interface board, which will be replaced free of charge if it fails during the warranty period. There is, of

```
Universal Pica
  10 Pitch       This is a sample of Universal Pica type face.
                 ABCDEFGHIJKLMNOPQRSTUVWXYZ !@#$% &*()  +  :"  ?
  Keyboard 1     abcdefghijklmnopqrstuvwxyz 1234567890 ‾=½¼;',./

                 ABCDEFGHIJKLMNOPQRSTUVWXYZ !£¨$[]^ç()  ´Ñ:`¡¿?
  Keyboard 2     abcdefghijklmnopqrstuvwxyz 1234567890 ‾éñ;',.|
```

Figure 1. Universal Pica typeface.

```
    Roma (Script)
      12 Pitch
                   This is a sample of Roma (script) type face.
      Keyboard 1   ABCDEFGHIJKLMNOPQRSTUVWXYZ !@#$%¢&*()  +  :",.?
                   abcdefghijklmnopqrstuvwxyz 1234567890 ‾=½¼;',./

      Keyboard 2   ABCDEFGHIJKLMNOPQRSTUVWXYZ !£¨$[]^ç()  ´Ñ:`¡¿?
                   abcdefghijklmnopqrstuvwxyz 1234567890 ‾éñ;',.|
```

Figure 2. Roma (script) typeface.

```
(DATE)

Dear (RECEIVER),

    I know that it has been (TIME) since I have written, but
(PROJECT) has really kept me busy.  I am quickly writing a short note
to let you (NAME) and (NAME) know that you have been in my thoughts.

                          etc.. etc..
```

Figure 3. The "forms fill-in" feature of a word processor using Roma 12 pitch.

course, a charge for out-of-warranty replacements. The interface of the Bytewriter is parallel, which is standard, but it uses a 20-pin connector. It plugs into the parallel port of the Model II no matter which way you plug it in—it is not marked, "This side up." When I first plugged it in, I got it upside down, and my word processor thought that it was printing, but I knew better. I turned the connector around, hit the ENTER key, and away it went.

The interconnecting cables are available for the following computers: TRS-80 Models I, II, III, and the Radio Shack Color Computer (with Microworks PI-80C interface); Apple II and III; Franklin Ace 1000 and 1200; Osborne I; Kaypro II; IBM PC; Atari 400 and 800; Commodore 64 and the VIC-20; TI-99/4A; NEC PC-8001; Zenith Z-100; Eagle II and IV; Victor 9000; HP-85; and Sanyo MBC-1000.

The Bytewriter has three switches to select Pica, Elite, or Mikron pitches (10, 12, or 15 characters per inch). It was shipped with a Universal Pica, 10-pitch print wheel (see figure 1). Yes, I had to have a wheel of a different pitch and type. After all, the feature is there. I was thinking of the "forms fill-in" feature of my word processor when I choose the Roma (script) 12 pitch (see figures 2 and 3).

Following the instructions to change print wheels is relatively easy. The ribbon cartridge does have to be removed and when it's replaced the ribbon only goes over the two guides in front. The back portion of the ribbon guide is for the correct tape.

The Bytewriter can be made to respond to a CR (carriage return) command in two ways. It has a switch so that it can provide a carriage return without a line feed or with a line feed. This feature is useful for extensive underlining and for some software that automatically inserts a line feed after every carriage return. Double spacing can be avoided by setting the switch to OFF. I immediately tested this feature because on some of my CP/M diskettes I did not execute the SETMISC & IOFREEZE utilities to suppress extra line feeds after CR's.

On the left side of the keyboard there is one more switch that gives you a second keyboard of printable characters. I used the typewriter manually and changed the switch when I wanted a character from the second keyboard. When the software is controlling the printing and the switch is on for keyboard 1, the desired characters from keyboard 2 can be inserted later (see figure 1). I left the space blank and inserted it later, manually. One time I did mess up a bit and that's when the memory erase of any of the last 10 characters came in handy. There is a lift-off or type-over correction ribbon built into the Bytewriter.

There are other printer/typewriters built on the Olivetti Praxis 30, but I found that Bytewriter was the most reasonable. The savings, by buying directly from Williams Laboratories, enabled me to purchase Radio Shack's SW-302 Parallel Printer Switch. I connected the cable from the switch box into the parallel port and connected the printers into the switch box. Nothing happened. It turned out that what is up for one connector, is down for another—that's why the cable wasn't marked "This side up." Now with a flip of a switch, I can watch my typewriter type at eight to 12 characters per second, or print a listing at 160 characters per second. I still use my line printer with my word processing program for hard copy correction sheets. It isn't until after the program is through marking my spelling errors and I've corrected them, that I switch over to the Bytewriter.

Of course, there is also the "regular typewriter" capability of the Bytewriter. If my computer goes down, I can use this typewriter/printer "manually." The speed really decreases then, but at least I wouldn't be at a complete standstill until the repairs are made.

I am extremely pleased with the beautiful copy that my Bytewriter produces. I have turned into a copy scrutinizer and I don't think there's a typewriter/printer that can surpass the Bytewriter in quality print.

Product:
Bytewriter

Manufacturer:
Williams Laboratories
125 Northview Road
Ithaca, N.Y. 14850
(607) 272-1132

System Requirements:
Any computer with a Parallel interface port

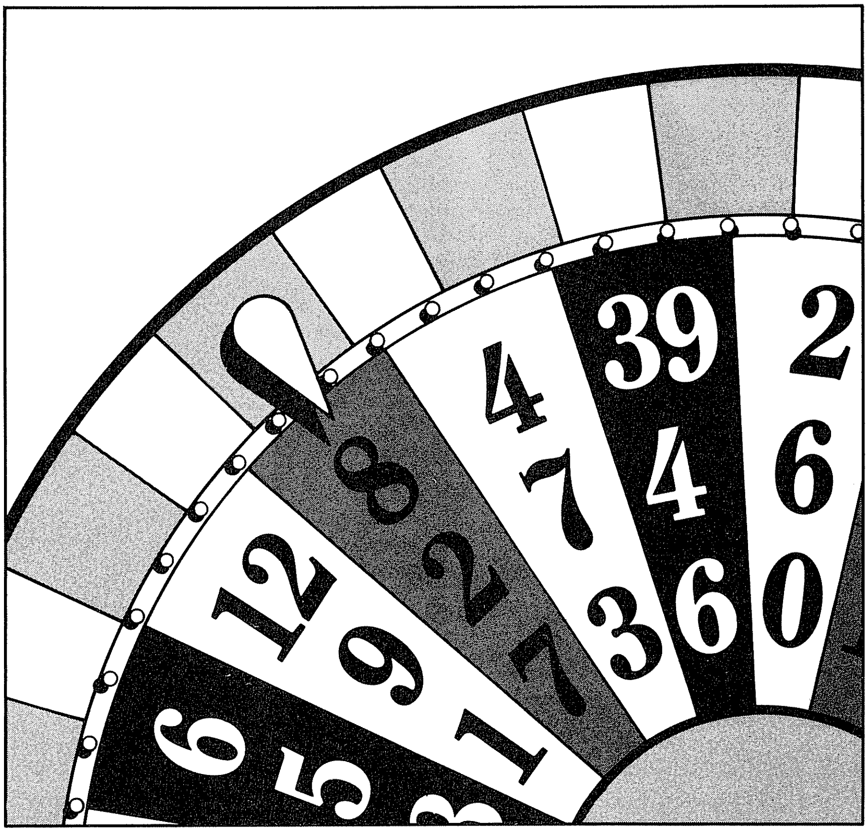
Special Requirements: None

Price: $495

A Random Number Generator for Pascal

This tutorial takes you through the simple math techniques of Pascal, then develops a no-frills Pascal program.

by Robert Athanasiou



Niklaus Wirth designed the Pascal language with the goal of developing good programming habits. These habits included logical program structure, considerable pre-program planning of algorithms, allocation of variables, and an easy-to-read program. The program's structure and function should be easily discernible from its source code.

The use of unlabeled GOSUBs and GOTOs that abounds in Basic programs is absent in Pascal. In fact, it has been claimed that the quality of a program is inversely proportional to the number of GOTO statements it contains. Pascal does have such a statement, but it is considered very bad form to use it... rather like cutting your spaghetti in an Italian restaurant.

The language is named after Blaise Pascal, a 17th century mathematician, physicist, and philosopher, who, at the age of 19, invented and patented a mechanical calculator that could add and subtract.

There are several implementations of Pascal for the TRS-80. The one I prefer is the Alcor System Version 2.0, which is sold under license by Radio Shack. It is a full implementation of the Pascal language with all its features intact. While Basic interpreters seem to be designed to be as different as possible among machines, the same is not true of Pascal. Pascal programs are far more transferable from one type of machine to another. This portability is only one of many good reasons to learn Pascal.

The first book I bought on Pascal was a game book written for the Apple computer. By the time I got to the second game program I realized that something was missing. While Basic has a handy built-in random number generator, Pascal does not. After hunting around in my library, I found an algorithm that claimed to produce pseudo-random numbers. Unfortunately it was a bust. The numbers were far from random and tended to repeat themselves at frequent intervals.

A quick trip to the local university library and some research on random number generating algorithms turned up some very interesting possibilities.

Since the numbers generated by any computer algorithm will eventually repeat their sequence, they are called pseudo-random numbers. For practical purposes, a long enough series can be produced so that the inevitable repetition is inconsequential.

The most commonly used method for generating these numbers is the residue method. Given three constants, A, C, and P, the residue method derives the $(i+1)$th number from the ith number by adding C to the product of X_i times A, and then taking the remainder, or resi-

```
{                     RANDOM NUMBER GENERATOR                         }
{                                                                     }
{                by Robert Athanasiou, PhD, MD                        }
{                         October 1983                                }
{              using ALCOR PASCAL for the TRS-80                      }
{                                                                     }
{                         adapted from                                }
{      Wirth's  ALGORITHMS + DATA STRUCTURES = PROGRAMS, 1976         }
{      Knuth's THE ART OF COMPUTER PROGRAMMING (Vol. 2), 1969         }
{      Forsythe, et al. COMPUTER METHODS FOR MATHEMATICAL             }
{                      COMPUTATIONS, 1977                             }
{*********************************************************************}

PROGRAM RANDY; { generates pseudo-random numbers }

   TYPE
      BYTE = 0..255;
   VAR
      SEED, RANGE : INTEGER;
      QUANTITY, NUMBER, K  : INTEGER;
      LIST : ARRAY [1..5000] OF INTEGER;
      MIN, SEC : BYTE;

{ The following are TRS-80 extensions of Pascal supplied with }
{ the ALCOR SYSTEMS Library.}

PROCEDURE CLEARSCREEN; EXTERNAL;
FUNCTION PEEK (ADDRESS : INTEGER) : BYTE; EXTERNAL;

{  ******          pseudo-random number generator          ******  }

FUNCTION RANDOM ( RNG : INTEGER ) : INTEGER;
   BEGIN
      SEED := ABS(((SEED * 3125) +  6917) MOD 32749);
      RANDOM := (SEED MOD RNG)+1
   END;                                        { of random function }

BEGIN  { The program to use the function to generate numbers  }

CLEARSCREEN;

WRITELN('Enter range for random numbers from 1..32748');
   READLN(RANGE);

WRITELN('Enter the quantity of numbers (up to 5,000) to be generated');
   READLN(QUANTITY);

SEED := PEEK(16918); { Sets Seed = to computer heart beat }

   FOR K := 1 TO QUANTITY DO    { loop to generate numbers }
      BEGIN
         NUMBER := RANDOM(RANGE); { random number created }
         LIST[K] := NUMBER;       { placed in array       }
         WRITE(LIST[K])           { write to screen       }
      END;                                   { For-Do loop }
END.                  { End of program; Note the period.}
```

Program listing

due, of division by P. For any number X_i, the algorithm is:

$X_{[i-1]} = ((A * X_{[i]}) + C)$ (modulo P)

The modulo function of Pascal doesn't exist in most Basics. It merely returns the remainder of a division. For example, in the equation 1 = 11 (modulo 2), 11 divided by two is five with a remainder of one.

A "seed" number, $X_{[0]}$, is needed to start the process. Each will produce its own sequence of pseudo-random numbers. If you use the same seed number each time, the number sequence will be the same. In most games, therefore, it is necessary to vary the seed number in some fashion so that the players will not get the same sequence of numbers each time the function is called.

Picking the numbers, A, C, and P is not an arbitrary process. Some heavy mathematics is involved and the interested reader is referred to Knuth's *The Art of Computer Programming: Seminumerical Algorithms (Vol. 2)* for an excellent source text. The number theory required to pick A, C, and P leads to the following criteria:

1. A should have four properties:
   a. A modulo 8 = 5
   b. $P/100 < A < (P - \sqrt{P})$
   c. The binary digits of A should have no obvious sequence (e.g., 110110110110...)
   d. A should be a prime number raised to an integral power
2. C should be an odd integer (preferably prime) such that $C/P \sim 0.21132$
3. P should be the largest integer prime that can fit in the computer word size.

A good choice for A is 3,125, which is 5^5, and in binary is 110000110101. It also meets the criterion that 3,125 modulo 8 = 5. That is, eight divided into 3,125 is 390 with a remainder of five.

C should be 6,921, but that number is not a prime. The nearest prime is 6,917, which is close enough.

The largest integer a Z-80 can handle is 32,767. The closest prime number smaller than that is 32,749. The random number algorithm then is:

$X_{[i+1]} = (3125 * X_{[i]} + 6917)$ modulo 32,749

X_i will range from zero to 32,748. This sequence will have a very long period before it repeats itself and the numbers will appear to be random. However, we rarely have a use for random integers in the range of 32,000. Most programs use random numbers that represent dice (1..12), cards (1..52), and other such limited sequences.

To change the range of pseudo-random numbers produced it is simply necessary to perform another modulo function. That is, if the range we want is from zero to 99, we merely use the expression X_i modulo 100. If zero is not desired, we add one to the result and the range is now one to 100.

The program listing shows this function in Pascal. Since it is written as a Pascal function, which is exactly like the DEF FN operator in Basic, we can call the RANDOM function at any point in a program, pass the upper bound of the desired range to it, and generate pseudo-random numbers to our heart's content.

Let's look at the compiled version of the program in the listing on a line-by-line basis. Line numbers are *not* used in Pascal. The compiler inserts them in the output listing as a convenience.

The first 12 lines are enclosed in brackets. This means they are comment lines just like the REM statements in Basic. Some Pascal implementations use asterisks to enclose remarks. Either notation works in the Alcor System. Comments are treated like blanks and can be inserted anywhere at all except in the middle of a number or word and between the parts of compound symbols (e.g., := or ←).

Blank lines and indentations are inserted to make the program more readable when you type in the source code. They are ignored by the compiler.

Line 14 contains the PROGRAM declaration that all Pascal programs must have. You can use any eight-character name for your program. This need not be the same as a disk file specification, but it's a good idea to keep the program name and the disk file spec similar so you'll remember which program does what.

A semicolon separates statements in compound lines. It serves the same function as a colon in Basic. Failure to include a semicolon is the most common error made in writing Pascal programs. Without a semicolon, the compiler doesn't know where a statement should end, and a whole series of compiler errors will be generated. As you scan the program, you will see that there are only a very few lines that don't need semicolons.

Since semicolons are used to separate statements, there is no semicolon required before the first statement or after the last statement in a sequence. You should *never* use a semicolon after an ELSE or before an END. Pascal, like Basic, is tolerant of null statements so an occasional extra semicolon will often not cause any trouble.

Blank spaces or delimiters must surround key words and there must be numbers on each side of a decimal point. The statement:

BEGIN X: = .5 + 1 END

is illegal because of the .5, which should be 0.5.

Lines 16 and 17 define a new type of variable called *byte*, and specify that its range will be from zero to 255. Note that a plain equal sign (=) is used here, whereas elsewhere the : = sign is used. The plain equal sign means that something is being defined or tested, such as IF A = B THEN...

Lines 18-22 are the variable declarations. Only two types of variables are used in this program: integer and byte. Other predefined Pascal variable types are real, character, Boolean, text, array, set, file, and record. String is a variable type in some implementations of Pascal including Alcor's.

In this section of the program, you must declare all the variables you intend to use on a global basis. That means that you have to think the program through before writing it. Unlike Basic, Pascal won't let you define or dimension variables as you go.

Lines 28 and 29 are external procedure and function calls. Each implementation of Pascal deviates from the standard in allowing you to call up some special functions and procedures from an external library. The Clearscreen procedure (Pascal does not distinguish among upper and lowercase letters) does just what it says. It is just like the CLS command in Basic. The Peek function requires an integer address (as specified in the parentheses), returns a value of the byte type, and is external to the program.

The function RANDOM is delared in line 33 of the program. The parentheses following the word RANDOM indicate that we must pass it an integer value which it will call "rng" and that it will return an integer value to the program.

The word BEGIN means that a group of statement are about to follow which will be grouped together until the compiler discovers the word END. These BEGIN-END groups are called statement bodies and must be included in every program block. It is possible to nest many of these in a single block.

In line 35, the variable (i.e., the memory location) called seed is assigned the value of the expression on the other side of the : = sign. Here the : = sign is used because we want to actually place (or assign) a value in a physical memory location. The function ABS is the same as it is in Basic and is used to prevent the generation of negative numbers if the value tries to exceed 32,767.

RANDOM is assigned a value in line 36. All functions, as distinct from procedures, must have a value assigned to the function name before the end of the statement body which occurs in line 37.

The actual program BEGINs in line 40. The Clearscreen procedure is called and then a prompt is written on the screen. WRITELN (pronounced write line) is the functional equivalent of PRINT in Basic. The text contained in the quote marks will be printed on one line of the screen. If we wanted to write something else on the same line we would use WRITE instead of WRITELN. WRITE is equivalent to PRINT; in Basic.

READLN(range); is the same as line INPUT R: in Basic. It accepts input from the keyboard until the ENTER key is pressed.

In line 50 we start the processs of generating pseudo-random numbers by assigning the variable seed (X_0) the number that is in memory location 16918. This location is the "heart beat" of the TRS-80 clock and will have a different (but not random) value each time we look at it. Since the exact second when this line is executed will vary from one program run to another, we will very rarely generate the same pseudo-random number sequence.

Line 52 starts a FOR loop. This is essentially the same as the FOR-NEXT loop in Basic. Pascal has many variations on this theme, however. There are REPEAT-UNTIL loops and WHILE-DO loops. Each is a bit different and this is what gives Pascal some of its awesome programming power.

After the BEGIN that starts the statement body of the FOR loop, we make our first call to the random number generator and assign the value it returns to the variable called NUMBER. NUMBER is in turn assigned to a spot in the array called LIST. We could have eliminated the step that uses NUMBER and assigned the random integer directly to LIST, but I'm trying to keep this simple. The next statement writes out the value of LIST. WRITE rather than WRITELN is used since we want several of the numbers on the same line. Pascal will format the output using default values just as Basic would.

The last line in the program is END. This is the mandatory last line in every Pascal program. A period *must* be used to tell the compiler that it has reached the end of the program.

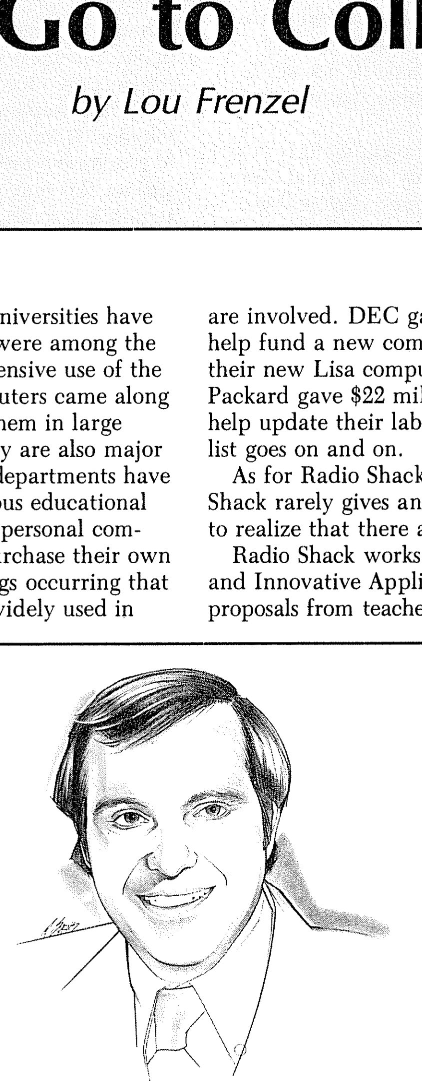
Robert Athanasiou is the Director of Emergency Services at Samaritan Hospital in New York. He has a Bachelor's degree in electrical engineering and a PhD in psychology.

PCs Go to College

by Lou Frenzel

Colleges and universities have always been heavy users of computers. They were among the first organizations in this country to make extensive use of the large mainframe computers. When minicomputers came along in the mid '60s, they were the first to adopt them in large numbers for teaching purposes. And now, they are also major users of personal computers. Most university departments have already acquired personal computers for various educational and administrative activities. The low cost of personal computers has also permitted many students to purchase their own units. Today there are several interesting things occurring that will make the personal computer even more widely used in colleges and universities.

During the past year, large computer corporations such as IBM, Digital Equipment Corporation (DEC), NCR, Wang, Apple, Honeywell, and even Radio Shack have been making donations of computers to selected colleges and universities. IBM, for example, just gave $50 million in cash and equipment to 20 universities to help graduate students familiarize themselves with the latest design and manufacturing techniques. Computer-aided design (CAD) and computer-aided manufacturing (CAM) are becoming more widely used in industry, but colleges have not been able to afford the expensive equipment to teach CAD/CAM. The donation will overcome this problem and help ensure a supply of knowledgeable engineers to industry.



Computer manufacturers are donating millions of dollars worth of equipment to colleges and everyone is benefiting— schools, students, and the corporations.

Just recently, IBM and DEC joined together to donate $50 million in equipment to the Massachusetts Institute of Technology. The purpose of the donation is to help MIT explore the use of computers as teaching tools. This covers a lot of ground, including computer-assisted instruction (CAI) and the use of the computer in problem solving. The goal is to attempt to develop a "computerized" curriculum.

While these were the largest recent donations, there are many smaller ones. They range from $100,000 to several million dollars. Most of the major manufacturers and colleges are involved. DEC gave $1.6 million to Boston University to help fund a new computer science program. Apple gave 50 of their new Lisa computers to Brown University. And Hewlett Packard gave $22 million in equipment to many universities to help update their lab equipment and teaching programs. The list goes on and on.

As for Radio Shack, it's a bit of a different story. Radio Shack rarely gives anything away, but they are smart enough to realize that there are benefits in the donations game.

Radio Shack works through a grants program called "Unique and Innovative Applications of Microcomputers." They solicit proposals from teachers and award equipment grants for worthy ideas. During the past year, they awarded a total of $113,508, with grants typically ranging from $3,000 to $12,000 each.

You can get more information on this program, which covers elementary schools through colleges, by writing Tandy TRS-80 Educational Grants Program, Radio Shack Education Division, 1400 One Tandy Center, Fort Worth, TX 76102.

As you might suspect, these donations are not totally altruistic and philanthropic. The corporations themselves stand to gain quite a bit.

First, the corporations pick up some tax benefits. They can write off all or part of their donations and enjoy the tax advantage. Current tax laws allow manufacturers to donate "scientific equipment" to colleges for basic research. Up to 10 percent of a company's taxable income may be donated and deducted. But this is probably not the major reason for making donations. The corporations have a much longer-range plan in mind.

What the manufacturers really want is for the students to become exposed to their computers. If the schools can't or won't buy them, the manufacturers will donate them. In this way, the students and the faculty will learn to use that type of computer. In many cases, this familiarity will lead to the

ultimate purchase of that type of computer when the student graduates. It's been known for a long time that students who use a particular brand of equipment in college will most likely go back to that brand when they must purchase equipment for themselves or their company later on. This is a long-term marketing strategy, but it works like a charm.

The problem with this approach, however, is that there is such rapid change in the personal computer field. I wonder whether the computers students use in college will actually be available after they graduate. Maybe the industry will settle down somewhat so that there can be greater stability and longer product life. But even if the same model is not available, no doubt the students will still remember the manufacturer. The effect can still be positive.

Another benefit is that once the donated equipment is in place, the schools will want and need to buy additional accessories, peripherals, software, and service. Such sales will go right to the manufacturer, and could even encourage the purchase of additional computers.

> The computer science programs in most colleges and universities are overloaded, but even though these colleges are turning out record numbers of graduates, there simply aren't enough to go around. Virtually every computer manufacturer and software firm is looking for qualified people.

In many colleges and universities where budgets are limited, students are not exposed to personal computers. Those schools with little or no computer capability will not graduate people who know how to use them. This is bad for business and industry as well as the students. The students won't be properly prepared for the jobs available. But more importantly, the colleges and universities may not be able to graduate the quantity and quality of graduates that the manufacturers themselves need so desperately.

There is actually a shortage of people who know about computers. The computer science programs in most colleges and universities are overloaded, but even though these colleges are turning out record numbers of graduates, there simply aren't enough to go around. Virtually every computer manufacturer and software firm is looking for qualified people. It is a fact that business growth is limited by the availability of qualified people.

If supplying personal computers to the schools allows them to graduate more students of higher quality, who should complain? The computer manufacturers will get the graduates that they need and the graduates themselves will find excellent jobs.

Another reason manufacturers donate computers is to replace obsolete equipment. Schools are just loaded with old out-of-date and irrelevant equipment. Their instruction suffers as a result. Students learn old technology rather than the new. This seriously affects the graduates and their employers. The manufacturers also like to use the schools as information sources about their products. In return for the equipment, the students and faculty can supply invaluable data on product performance, limitations, problems, and needs that will ultimately help the manufacturers revise their equipment or design better products to become more competitive.

While it may appear selfish for the manufacturers to give computers to the schools, overall it looks like a good deal for everyone.

Students already have to meet a number of major prerequisites to get into college. They have to be high school gradutes, their grades must be of a certain level, they must usually pass an entrance examination of some kind (SAT, etc.), and often they are personally screened through an interview process. It's not that easy to get into a major university these days. And even when they do get in, students are faced with the problem of financing their stay.

Today, many colleges and universities are making it even tougher. A number of schools are requiring entering freshmen to purchase a personal computer when they enroll. On the surface this may look like a burden, but this computer prerequisite is really a benefit to both the school and the student. And industry will ultimately benefit from having better trained graduates.

I can remember talking to an MIT professor of computer science in the late 1970s. He predicted that sooner or later engineering colleges and universities would require each student to have a computer. I'm not too sure I believed him at the time, but I thought that it was a novel idea that had some merit. His prediction has certainly come true. Today there are quite a few colleges and universities teaching scientific and engineering subjects that have the computer prerequisite.

When you stop and think about it, scientists and engineers are big users of computers. Almost everything they do requires some kind of mathematical operation, computation, or tabulation. Once a student learns the concepts with manual computations, it's nice to be able to speed up and simplify those operations with some kind of computer. Calculators are still widely used, of course, but they can't compete with computers that can do so much more.

In the "olden days," students who needed to use a computer would go to the main computer center, punch out their program in Fortran on cards, then have the big mainframe computer run it on a batch basis. While this worked, it was incredibly frustrating and time consuming. Many times the program had bugs and wouldn't work the first time. This meant going back to debug the program, punching new cards, and having the program rerun. It often took many days, sometimes even weeks, to get a program written, debugged, and the results back. When the number of students running programs grew, the problem became insurmountable. Students were spending long nights and early mornings in the computer center competing for key punch access and computer time to get their programs processed.

While more capable time-sharing computers have relieved this somewhat, and while many departments have their own mini and microcomputers now, the burden is still there. Once the faculty and students get a taste for what a computer can do, they want to use it more. The demand has increased tremendously. It has taken a lot of drudgery out of scientific, mathematical, and engineering work. It permits the students to work at a higher level and save time. Personal computers really make sense for college.

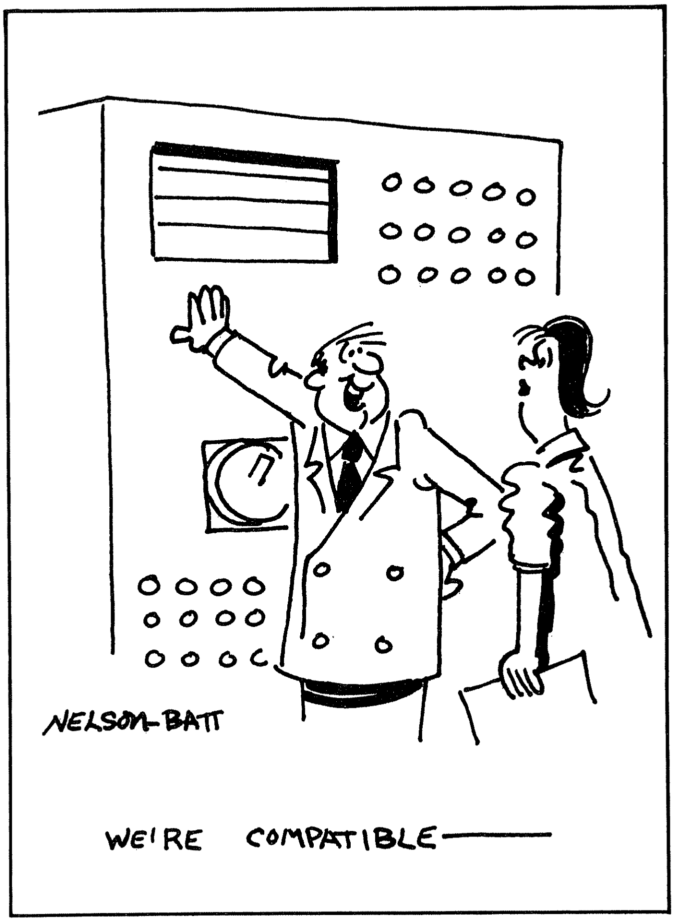
By requiring every student to have a particular type of computer, the school is able to structure courses and a complete curriculum that makes use of this machine. The students learn basic computer literacy as well as programming. They use the computer for everyday problem solving, and large numbers of students use the word processing capability of the computer to turn out professional looking papers and reports. By adding a modem, the student can use the computer as a terminal to access the more powerful university computer from the comfort of his or her own dorm room.

What kind of computers are being required? Just about every brand is represented, including the Atari 800, the IBM PC, the Zenith Z100, the Osborne and Kaypro portables, and even the new IBM PCjr. The forthcoming Apple 32 (or Mac Intosh) has also been selected. Recently, Dallas Baptist College selected the TRS-80 Model 100 portable (see article in this issue). Students can even use this one in class.

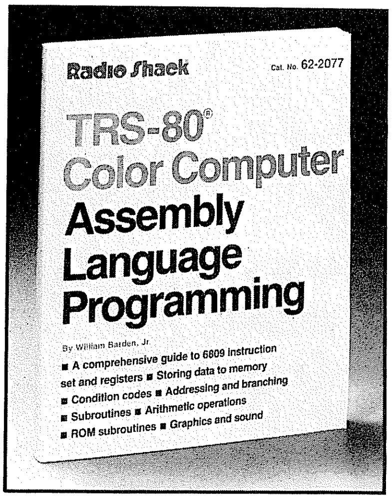
In most cases, the schools are acting as distributors for the computers. They get good "educational" discounts and pass along the low prices to students. Many even spread the students' payments out over the four-year period to help relieve the financial burden. When the student graduates, he or she will own the machine.

The computer prerequisite is expected to expand into other schools over the years. A personal computer is certainly an affordable thing today and many students already have them. The students love them and are looking for some justification to get one anyway. It all sounds like a good arrangement to me because everyone involved—the students, the schools, and the manufacturers—benefits from the deal.

Lou Frenzel is a full-time freelance writer and has authored many articles, columns, books, and self-study courses on microcomputer and electronic subjects. He is author of the best-selling book Crash Course in Microcomputers *(Sams). Previously, he was Senior Vice President for Product Development at McGraw-Hill Continuing Education Center in Washington, D.C., and Vice President at Heath Company, where he managed the Heathkit/Zenith education and personal computer product lines.*



Long-awaited Assembly Language Book Released



TRS-80 Color Computer Assembly Language Programming
William Barden, Jr.
Radio Shack
Cat. No. 62-2077
Reviewed by McAlister J. Merchant, Jr.

It has been slow in coming and eagerly anticipated. Finally, Radio Shack has released the Color Computer Assembly Language programming book by Bill Barden. The long wait was worth it.

For more than a year, I have been making sporadic attempts to learn Assembly language programming for the 6809. My shelf has been growing heavier every month with all manner of literature related to computing in general and the Color Computer in particular. Each of these books has its place and its charm, but not one begins to approach the cogency of the new Bill Barden book. The book is, by any measure, a unique volume of creative journalism and inspired instructional tactics.

There are two areas of special note: the book's structure and its use of examples. The book is uniquely designed. Every chapter starts with an edited index that displays the progress through the list of Assembly language concepts and mnemonics by showing which topics have been covered in preceding chapters, which items are to be examined and exampled in the current chapter, and which are to be covered in subsequent chapters. Within each chapter, helpful but extrinsic information is set apart from the principal text in graphic blocks labeled "Hints and Kinks." I found this particularly helpful when rereading portions of chapters because it made scanning for information easier.

There are many examples included in the book. They are short and easy to key-in. Each example is clearly illustrative of just the immediate material, and they are clearly and completely explained.

I have but one serious complaint about this book. It needed one more pass through the hands of a competent proofreader. There are so many typographical errors and omissions in the book that I'm certain Bill Barden was embarrassed at its publication even if Radio Shack wasn't. It's a considerable credit to the talents of Barden that the errors are soon discovered, and, with his clear explanation of what should be happening, they are easily corrected (you can test this on page 41 if you want to see what I mean).

Despite the typos, I recommend this Assembly language programming book above all others (even, perhaps, for other microprocessors). You will not find another book that will teach you as much about Assembly language programming as *TRS-80 Color Computer Assembly Language Programming.*
296 pages $6.95 1983

Basic Disk I/O Faster and Better
Lewis Rosenfelder
IJG Publishing Inc.
ISBN 0-936200-25-1
Reviewed by Harry Avant

Lewis Rosenfelder's latest book, *Basic Disk I/O Faster and Better & Other Mysteries*, is an excellent follow-up to his previous best seller, *Basic Faster and Better*. This latest book has more than 400 pages of information covering every aspect of disk I/O routines on the TRS-80 computer.

The first three chapters are concerned with the elementary aspects of using a TRS-80. Because the experienced user will be familiar with the subject, these portions will probably be skipped by all but new computer users.

The real value of this book begins with chapter four, where concepts related to sequential files are introduced. The next chapter introduces the random file process, which is usually the stopping place for most computer users. But in this book, it is only the beginning.

If you had always thought that a B-Tree was something that would most likely be found in a forest, Rosenfelder is going to change your mind. ISAM, or Index Sequential Accessing Method, is also given a good going over. In spite of the formidable sounding names, and even more difficult concepts, the material is presented in a step-by-step sequence, which makes for easy reading.

All of the material contained in the book reflects writing by an author who knows the concepts of disk I/O and under-

stands the difficulty of presenting the material in a readable manner. Good examples for each of the file types and search methods are included. The program segments in the text are also available on disk, which would be a wise investment as it will save you the task of typing in the few hundred lines of code that are used as examples.

By the time you have mastered the contents of this book, you will be able to write disk file programs that are quite fast and very free of errors. This is not a book that can be read in one sitting, nor was it intended to be. It is a complete text on the subject of disk input and output and is not a book that will soon be outdated.

In addition to the traditional presentation, the author has included little excerpts from his own "on the job" programming, which gives the reader insight as to why some of the methods presented are used.

This book also points out the modifications that are needed to use the various examples on the Model II, or on the Model 12 or 16 running under TRSDOS 2.0 or 2.0a.

430 Pages $29.95 1983

Using the Radio Shack TRS-80 in Your Home
Kenniston W. Lord, Jr.
Van Nostrand Reinhold Co.
ISBN 0-442-26079-2
Reviewed by James C. Graves, Jr.

Education, scheduling, money management, and games illustrate the kinds of applications advocated by this book. It teaches a logical sequence for writing programs—the construction process proceeds in a step-by-step manner, going from simple to complex. The programs will run on the Model I with 4 KB of RAM and a cassette storage system.

The first three chapters give a short background of computing, a look at future uses, and fundamentals of Basic programming. Since the publication date is 1981, much of the look at the future only serves to reinforce the fact that computers are now part of our daily environment. The major portion of these chapters teaches Basic, including how to use loops, subroutines, arrays, and other functions. Modules are developed for inclusion in the programs to come. Routines for sequential table searching, exchange sorting, and block sorting show effective ways to manipulate data. Building variable tables is also explained.

The remainder of the book deals with building and using programs. A section on graphics shows how to draw floor plans and make bar charts. For the home, shopping lists, inventory tracking, calculating energy consumption, and auto miles-per-gallon and maintenance recording prove to be valuable jobs for the Model I. Game programs furnish entertainment.

The book concludes with a personal accounts payable program. The program is put together in phases. Phase one is data collection, and phase two is a do-everything module that gives status reports and does forecasting, among other things. Finally, the check writing phase (which does not actually write checks out) assigns check numbers, tracks the checkbook balance, and prepares a ledger entry.

Readers with limited knowledge in programming will not be intimidated by this book. The reading is easy and the programs are fully explained. Not only are the programs beneficial, but they are also stepping stones to more advanced uses of your computer.

457 pages $15.50 1981

Crosswires? When you have crossed wires you have a definite exchange of information. Sometimes this information exchange is a valuable one, other times, not. But with the proliferation of computer communications throughout the country, we are confident that this column will be a valuable service to our readers.

Every month we will have a listing of Bulletin Board Systems throughout the United States and perhaps the world. Our technical staff has personally contacted each of these Bulletin Boards, so we're reasonably sure that the numbers are current. You can expect this list to grow each month as the System Operators start to send in their information.

Crosswires will also have a listing of TRS-80 computer clubs and users groups. These organizations present a great opportunity for information exchange. Take advantage of the groups in your area—they're where the real expertise is.

If you are a Bulletin Board System Operator, please send a postcard with the name of your system, the modem phone number, the hours, the baud rate, the program type you are using (TBBS, Green Machine, etc.), and any pertinent information that makes your system different. Also, please include your name, address, and voice telephone number (in case we need to contact you regarding the system).

We would like to ask each of the clubs and users groups to please send us the name, street address, and telephone number of the person to contact for information. Please include the meeting hours and any pertinent information on the club or group.

Send this information to:
Attn: Crosswires
Computer User
16704 Marquardt Ave.
Cerritos, CA 90701

BULLETIN BOARDS

Bulletin Board Name	Telephone Number	Baud Rate	Operating Hours
CFONJ	201-486-2956	300-1200	24
Forum 80 Monmouth	201-528-6623	300	24
Photo-80	201-790-6795	300-1200	24
Dial-Your-Match	206-256-6624	300	24
Forum 80 Seattle	206-723-3282	300	24
JCTS	206-883-0403	300	24
Mini-Bin	206-762-5141	300	24
PMS Software	206-486-2368	300	24
US-Online	206-692-8408	300	24
Rainbow Connection	212-441-3755	300-1200	24
Data Exchange	213-926-9553	300-1200	24
Poor Richards BBS	213-502-0907	300	24
Daltrug	214-289-1386	300	24
Microserve	214-769-3036	300-1200	24
Bullet-80	216-729-2769	300-1200	24
NASA Get Away Special	301-344-9156	300-1200	24
Apparat, Inc.	303-741-4071	300-1200	24
C/S CCS	303-598-4500	300	24
Northstar	303-444-7231	300	24
Old Colorado City	303-632-3391	300-1200	24
Pinecliff RMPM	303-642-3034	300-1200	24
RBBS	303-499-9169	300-1200	24
TBBS #1 Headquarters	303-690-4566	300-1200	24
UFOnet of Colorado	303-278-4244	300-1200	24
Corsair	305-968-8653	300-1200	24
Infoex-80	305-683-6044	300	24
The Notebook	305-686-4862	300	24
Winter Connection	305-894-1886	300	24
Aurora Computer	312-897-9037	300	24
C.A.M.S.	312-927-1020	300-1200	24
Co-Operative Computers	312-359-9450	300	24
L.A.M.S.	312-740-9128	300	24
Barton Data Line	313-662-8303	300-1200	24
Bullet-80	313-683-5076	300	24
Central Processing Unit	313-547-7903	300	24
Nautilis	316-365-7559	300-1200	24
Salt City Swapper	316-663-3600	300-1200	24
HBBS	405-848-9329	300-1200	24
Helena Valley RBBS	406-443-2768	300-1200	24
Most Significant Byte	406-256-3813	300	24
Serious Users Exchange	414-327-6010	300	24
Aardwolf Express	415-651-4147	300-1200	24
Sunrise Computer	415-452-0350	300	24
Little Connection	501-372-0576	300	24
LNW Southeast	504-291-4331	300-1200	24
TBBS Baton-Rouge	504-926-0181	300-1200	24
TBBS of Austin	512-385-1102	300	24
The Datapoint	516-467-6545	300-1200	24
Bable Net	517-485-6232	300-1200	24
Compunet	517-339-3367	300-1200	24
MMS-Color/80	517-484-2448	300	24
Blax-80	602-952-1382	300	24
Green Machine of Yuma AZ	602-726-7533	300-1200	24
Hub Graphics	617-569-9140	300	24
The Outpost	617-259-0181	300-1200	24
Comnet-80	702-870-9986	300-1200	24
Grape Line BBS	707-538-9124	300	24
The Wine Country	707-257-1101	300	24
Color Corner BBS	714-350-2668	300	24
Computer Corner	714-653-9426	300	24
Growth Net	714-359-1586	300	24
Kandy Shack	714-354-8004	300-1200	24
Kandy Shack	714-877-2505	300-1200	24
Moreno Valley Green Machine	714-924-2229	300	24
OCTUG	714-530-8226	300-1200	24
TBBS	714-824-2129	300	24
TBBS Wausau	715-352-2093	300	24
Mouse-Net	802-862-7023	300-1200	24
Burg Board	806-374-9711	300	24
Tele-Med-Comm	817-526-5915	300	24

This listing continued from CU Nov 83

MISSISSIPPI

GUM TREE COMPUTER CLUB TRS-80
2609 PEMBERTON
TUPELO, MS 38801

NORTH CAROLINA

GREENSBORO TRS-80 USERS GROUP
PO BOX 10014
GREENSBORO, NC 27404

KIDS COCO CLUB
914 ALBANY CT.
RALEIGH, NC 27609
NATIONAL COMPUTER SOCIETY
PO BOX 41205
FAYETTEVILLE, NC 28304
TRIAD AMATEUR COMPUTER SOCIETY
5515 W. MARKET ST., APT 801
GREENSBORO, NC 27409
WILMINGTON TRS-80 USERS GROUP
2031 CHESTNUT ST.
WILMINGTON, NC 28405
WILSON TRS-80 USERS GROUP
PO BOX 336
LUCAMA, NC 27851

NEBRASKA

COMPUSERS
P.O. BOX 2064
HASTINGS, NE 68901
LNW USERS
4345 MANCHESTER ROAD
GRAND ISLAND, NE 68801

NEW JERSEY

AMATEUR COMPUTER GROUP OF NJ
P.O. BOX 319
SOUTH BOUND BROOK, NJ 08880
CENTRAL JERSEY COMPUTER USER CLUB
RD#1, BOX 147
HOPEWELL, NJ 08525
COMPUTER AMATEURS OF SOUTH JERSEY
1100 NORTH KINGS HIGHWAY
CHERRY HILL, NJ 08034
COMPUTER CLUB OF OCEAN COUNTY
1319 GEORGIAN TERRACE
LAKEWOOD, NJ 08701
NORTH JERSEY COMPUTER USERS GROUP
P.O. BOX 382
SPARTA, NJ 07871
NORTHERN NJ COMPUTER CLUB
6 BRYSON ROAD
FAIRLAWN, NJ 07410
TRS-80 USERS GROUP OF CHERRY HILL
416 VIRGINIA DRIVE
BROWNS MILLS, NJ 08015
TRS-80 USERS GROUP OF CHERRY HILL
469 WESTMINSTER ROAD
WENONAH, NJ 08090
TRS-80 USERS GROUP OF MANMOUTH
80 BINGHAM AVENUE
RUMSON, NJ 07760

NEVADA

LAS VEGAS TRS-80 USERS GROUP
1941 SHANER LANE
LAS VEGAS, NV 89115
LAS VEGAS TRS-80 USERS GROUP
1941 SHANER LANE
LAS VEGAS, NV 89115

NEW YORK

COLOR COMPUTER CLUB OF NEW YORK
347 W. 48TH ST.
NEW YORK, NY 10036
CP/M USERS GROUP
345 E. 86TH ST.
NEW YORK, NY 10028
DIGITAL GROUP INDEPENDENT USERS GRP
PO BOX 316
WOODMERE, NY 11598
EDP NETWORK
PO BOX 41
MASONVILLE, NY 13804
KINGS BYTE
1063 E. 84 ST.
BROOKLYN, NY 11236
LONG ISLAND COMPUTER ASSOC., INC
PO BOX 71
HICKSVILLE, NY 11801
METRO TRS-80 USERS GROUP
310 WEST 106 ST. 15D
NEW YORK, NY 10025
MICRO COMPUTER BUSINESS USERS GROUP
161 W. 75 ST.
NEW YORK, NY 10023
NATIONAL ADV. STOCK/COMMODITY GROUP
LOCK BOX 407
NEW YORK, NY 11362
NEW YORK AMATEUR COMPUTER CLUB
PO BOX 106, CHURCH ST. STATION
NEW YORK, NY 10008
NORTH COUNTY TRS-80 2 & 16 CLUB
24 WOODLAND TERRACE
LAKE PLACID, NY 12946
ROCHESTER AREA MICROCOMPUTER SOC.
PO BOX 90808
ROCHESTER, NY 14609
ROCHESTER TRS-80 USERS GROUP
91 VALLEY ST.
ROCHESTER, NY 14612
SUFFOLK COUNTY COMPUTER ASSOCIATION
RT. 1, BOX 244, MILL RD.
YAPHANK, NY 11980
THE DATAMERICA COMPUTER USERS GROUP
312 E. 84TH ST. #1A
NEW YORK, NY 10028
THE ROCHESTER S-80 COMPUTER CLUB
PO BOX 7476
ROCHESTER, NY 14615
TRS-80 USERS GROUP OF CENTRAL NY
2362 EAST LAKE RD.
SKANEATELES, NY 13152

OHIO

ACSCO-SWAP
BOX 28606
COLUMBUS, OH 43228
AMATEUR COMP. SOCIETY OF CENTRAL OH
PO BOX 28606
COLUMBUS, OH 43228
CINCINNATI TRS-80 USERS GROUP
3211 MARYLAND AVE.
CINCINNATI, OH 45204
CINCINNATI TRS-80 USERS GROUP
5380 BAHAMA TERRACE APT 8
CINCINNATI, OH 45223
COLUMBUS & CEN. OH COLOR COMP. CLUB
244 NORTH PARKWAY DR.
DELAWARE, OH 43015
COLUMBUS MAX80 USERS
BOX 28355
COLUMBUS, OH 43228
COLUMBUS TRS-80 COLOR COMPUTER USER
2153 LEAH LN.
REYNOLDSBURG, OH 43068
THE COLOR COMPUTER CLUB
PO BOX 478
CANFIELD, OH 44406
TRS-80 COLOR COMPUTER USERS GROUP
527 MALVERN DR.
PAINSVILLE, OH 44077
TUSCARAWAS CNTY PERS COMPUTER CLUB
231 5TH DR. NW
NEW PHILADELPHIA, OH 44663

OKLAHOMA

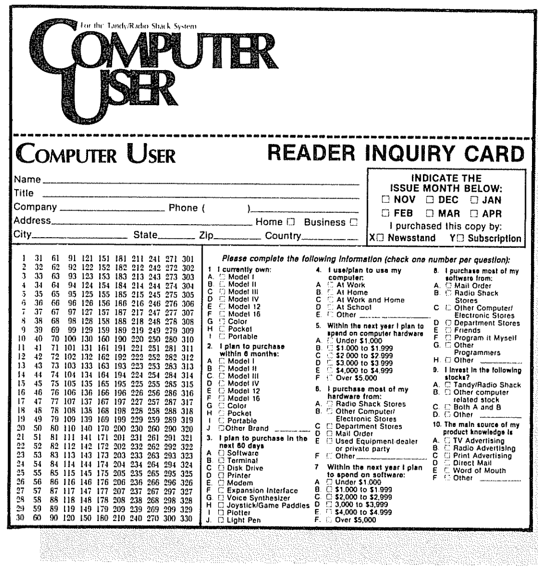
TULSA COMPUTER SOCIETY
PO BOX 1133
TULSA, OK 74101

List to be continued in future issues.

Compare products before you buy. . . use our *Reader Service Card to obtain the latest product information.*

To receive further information on new products listed in this section—as well as those displayed in advertisements throughout the magazine—please use the Reader Service Card found next to the inside back cover. Fill it out completely, circling the numbers that correspond to the items of interest. Mail it promptly. Requests are then forwarded on to the respective companies, who will send current information to you by return mail.

Note: The *Computer User* editorial staff assumes no responsibility for the claims detailed in the following product announcements. The information is supplied directly by the product manufacturers, and is not based on staff evaluations.



For the Tandy/Radio Shack System

COMPUTER USER

COMPUTER USER READER INQUIRY CARD

Name ____
Title ____
Company ____ Phone ( ) ____
Address ____ Home ☐ Business ☐
City ____ State ____ Zip ____ Country ____

INDICATE THE ISSUE MONTH BELOW:
☐ NOV ☐ DEC ☐ JAN
☐ FEB ☐ MAR ☐ APR
I purchased this copy by:
X☐ Newsstand Y☐ Subscription

1	31	61	91	121	151	181	211	241	271	301
2	32	62	92	122	152	182	212	242	272	302
3	33	63	93	123	153	183	213	243	273	303
4	34	64	94	124	154	184	214	244	274	304
5	35	65	95	125	155	185	215	245	275	305
6	36	66	96	126	156	186	216	246	276	306
7	37	67	97	127	157	187	217	247	277	307
8	38	68	98	128	158	188	218	248	278	308
9	39	69	99	129	159	189	219	249	279	309
10	40	70	100	130	160	190	220	250	280	310
11	41	71	101	131	161	191	221	251	281	311
12	42	72	102	132	162	192	222	252	282	312
13	43	73	103	133	163	193	223	253	283	313
14	44	74	104	134	164	194	224	254	284	314
15	45	75	105	135	165	195	225	255	285	315
16	46	76	106	136	166	196	226	256	286	316
17	47	77	107	137	167	197	227	257	287	317
18	48	78	108	138	168	198	228	258	288	318
19	49	79	109	139	169	199	229	259	289	319
20	50	80	110	140	170	200	230	260	290	320
21	51	81	111	141	171	201	231	261	291	321
22	52	82	112	142	172	202	232	262	292	322
23	53	83	113	143	173	203	233	263	293	323
24	54	84	114	144	174	204	234	264	294	324
25	55	85	115	145	175	205	235	265	295	325
26	56	86	116	146	176	206	236	266	296	326
27	57	87	117	147	177	207	237	267	297	327
28	58	88	118	148	178	208	238	268	298	328
29	59	89	119	149	179	209	239	269	299	329
30	60	90	120	150	180	210	240	270	300	330

Please complete the following information (check one number per question):

1. I currently own:
A. ☐ Model I
B. ☐ Model II
C ☐ Model III
D ☐ Model IV
E ☐ Model 12
F ☐ Model 16
G ☐ Color
H ☐ Pocket
I ☐ Portable

2. I plan to purchase within 6 months:
A ☐ Model I
B ☐ Model II
C. ☐ Model III
D ☐ Model IV
E ☐ Model 12
F ☐ Model 16
G ☐ Color
H ☐ Pocket
I ☐ Portable
J ☐ Other Brand ____

3. I plan to purchase in the next 60 days
A ☐ Software
B. ☐ Terminal
C ☐ Disk Drive
D ☐ Printer
E. ☐ Modem
F ☐ Expansion Interface
G. ☐ Voice Synthesizer
H ☐ Joystick/Game Paddles
I ☐ Plotter
J. ☐ Light Pen

4. I use/plan to use my computer:
A ☐ At Work
B. ☐ At Home
C ☐ At Work and Home
D ☐ At School
E. ☐ Other ____

5. Within the next year I plan to spend on computer hardware
A. ☐ Under $1,000
B. ☐ $1,000 to $1,999
C ☐ $2,000 to $2,999
D ☐ $3,000 to $3,999
E ☐ $4,000 to $4,999
F ☐ Over $5,000

6. I purchase most of my hardware from:
A. ☐ Radio Shack Stores
B. ☐ Other Computer/ Electronic Stores
C ☐ Department Stores
D ☐ Mail Order
E ☐ Used Equipment-dealer or private party
F ☐ Other ____

7 Within the next year I plan to spend on software:
A ☐ Under $1,000
B. ☐ $1,000 to $1,999
C. ☐ $2,000 to $2,999
D ☐ 3,000 to $3,999
E. ☐ $4,000 to $4,999
F. ☐ Over $5,000

8. I purchase most of my software from:
A. ☐ Mail Order
B. ☐ Radio Shack Stores
C ☐ Other Computer/ Electronic Stores
D ☐ Department Stores
E ☐ Friends
F ☐ Program it Myself
G. ☐ Other Programmers
H. ☐ Other ____

9. I invest in the following stocks?
A. ☐ Tandy/Radio Shack
B. ☐ Other computer related stock
C. ☐ Both A and B
D. ☐ Other ____

10. The main source of my product knowledge is
A. ☐ TV Advertising
B. ☐ Radio Advertising
C ☐ Print Advertising
D ☐ Direct Mail
E ☐ Word of Mouth
F ☐ Other ____

Personal/Home

Smal-LDOS 5.1 operating system

Logical Systems, Inc. has now made available to the general public the Smal-LDOS operating system. Smal-LDOS is a sophisticated Z-80 operating system for the TRS-80 line of microcomputer systems. Previously, this low priced subset of the LDOS operating system had only been available to OEM hardware or software distributors. Now this system costs less than half the price of the full LDOS system, and still contains many of the key features of the complete "large" LDOS. Smal-LDOS contains all of the popular LDOS commands and many valuable functions, including Job Control Language, keyboard type-ahead, and a powerful printer forms controller. An advanced version of TRS-80 Disk Basic (LBasic) is included, providing many features indispensable to the serious user. Smal-LDOS is available for the Models I/III/4 for $59 plus $4 shipping charge.
Logical Systems, Inc.; Milwaukee, WI
CIRCLE NO. 200

Interactive reading program

Magic Carpet, a new interactive fiction series on cassette, will provide young readers with the excitement of a voyage by magic carpet to an exotic country. From time to time the reader is asked to choose a course of action—enter the tower or the marketplace, go into the alley or the street? Each choice of action affects the outcome of the tale. When the story ends, the tape can be rewound and the story read again, choosing other alternatives. The story can be read again and again. Magic Carpet (26-1919) is available for $9.95 and is meant for use with a 16 KB Model III with Model III Basic, or with the Model 4 in Model III mode.
Radio Shack; Ft. Worth, TX
CIRCLE NO. 201

CP/M for Model 4

Montezuma Micro has announced the first known implementation of Digital Research's CP/M 2.2 operating system for the Radio Shack Model 4 computer. The system functions completely in either a 64 KB or a 128 KB Model 4. Users who have 128 KB can utilize the MemLink program, which allows the upper 64 KB of memory to be used as a RAM-based disk drive. Also built into the system is a configuration program for 35, 40, 77, and 80-track single- or double-sided drives, and a format utility that will build over 52 different diskette formats. Other features are full utilization of the function keys, 4 MHz clock speed, and ADM-3A emulation along with reverse video. The system code has been compressed to occupy only the first two tracks of the diskette, leaving 170 KB remaining for data storage. The operating system comes complete with all utilities and a 300-page user manual for $199.95.
Montezuma Micro; Dallas, TX
CIRCLE NO. 202

Spreadsheet program

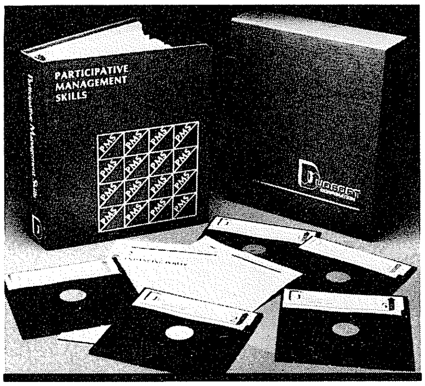
Target PlannerCalc is a new software package that offers the versatility of higher-priced electronic spreadsheets for under $100. Suitable for use with Model 4, Model III, and Model I computers, PlannerCalc's easy-to-use design minimizes the need for an extensive user's manual. The program allows creation of financial models to meet individual needs by entering and processing data by column, row, or individual location. A wide variety of mathematical operations, including conditional commands, are also offered. PlannerCalc is a spreadsheet program that uses plain English formulas instead of more complex algebraic formulas. For example, an obscure equation like "A1 = 500" can be written as "LINE 1 WIDGETS = 500." A "Help" screen that details an operation is available by pressing the question mark key. PlannerCalc (26-1512) is available for $99.95.
Radio Shack; Ft. Worth, TX
CIRCLE NO. 203

Personal/Business

Bookkeeping integration program

The Order Entry System can integrate Micro Architect Inc.'s INV-X (inventory control system) and AR (accounts receivable system). It can print invoices and update the inventory file at the same time, saving the user hours of repeated

entry. OE gets the name and address information from the AR file, and prints the invoice header information. You can enter multiple inventory items one by one. The inventory item will be checked and updated on-line. After all items are entered, sales tax and shipping charges will be printed with a grand total, and a sales transaction will be added to the AR file. Order Entry requires a TRS-80 Model II (or Model 12 or Model 16 under the Model II mode) with two disk drives (three disk drives or one hard disk recommended). The price is $99. The entire system with the inventory system and the accounts receivable is $479.
MicroArchitect, Inc.; Burlington, MA
CIRCLE NO. 206



Management skills software

The Participative Management Skills (PMS) package is a set of interactive tutorial lessons designed to help improve employee-management communications and to reverse trends toward declining productivity in business and industry. An instructional text and all necessary computer software are included in the PMS package. Specific subject matter covers the benefits of participative decision making; positive ways to use power within the corporate structure; how to facilitate team communication; understanding motivational dvnamics: and achieving concensus. PMS is available for users of TRS-80, Apple II, and IBM PC computers.
Duosoft Corp; Savoy, IL
CIRCLE NO. 207

Built-in hard disk Model 16B

The new TRS-80 256 KB one-disk Model 16B Microcomputer with 15 MB Hard Disk comes with a self-contained 1.25 MB 8" floppy disk drive and a 15 MB hard disk drive. This computer system allows access up to 15 million characters of information on hard disk. One more 12 MB external hard disk drive can be added for a total of about 9,000 pages of hard disk storage. The Model 16B with Hard Disk comes with the TRS-Xenix multi-user operating system. The computer, two low-cost DT-1 data terminals ($699 each) and connecting cables are the only equipment needed for multi-user operation. Using the TRS-Xenix operating system, all data in the Model 16B with Hard Disk can be stored on the hard disk. Files can be shared by all users in the system, or restricted to certain users only. Peripherals attached to the Model 16B with Hard Disk—printers, plotters, and modems—can be shared by all system users for maximum savings. TRS-Xenix is supported by a variety of multi-user software. The Model 16B can also run all Model II and Model 12 programs (for single-user operation only). The Model 16B's dual-processor design features Z-80A and state-of-the-art MC68000 16/32-bit microprocessors to permit the high operating speeds and large memory capacity required by many business users. The Model 16B with 15 MB Hard Disk is available for $6,999.
Radio Shack; Ft. Worth, TX
CIRCLE NO. 208

Pockets/Portables

Traveling appointment manager

The Traveling Appointment Manager is available for the Radio Shack Model 100 and the NEC PC-8201 notebook-size computers. The Appointment Manager is one of eight specially designed products within the Business Manager Series. The other seven are the Traveling Time Manager; the Traveling Expense Manager; the Traveling Sales Manager; the Traveling Project Manager; the Traveling Tax Manager; the Traveling Accountant Manager; and the Traveling Communicator. The Appointment Manager provides a complete system to easily manage your daily activities. You may schedule your appointments as far in advance as desired, until the year 2000. The Appointment Manager may also be used with most office or home computer systems using The Traveling Communicator package. The Traveling Communicator will allow for the formatting and transmission of appointment records to be transmitted via the RS-232 or telephone modem ports. The Traveling Appointment Manager is $59.95 and includes a tutorial notebook and an audio voice tutorial on cassette tape featuring "The Traveling Professor," who provides instructions on using The Traveling Appointment Manager and tips on getting the most out of your notebook computer.
Traveling Software, Inc.; Seattle, WA
CIRCLE NO. 209

Time Organizer
Pocket Organizer is a personal organizing aid for use with the PC-2. The ready-to-run software package is an organizing aid that stores a list or "queue" of things to do. Instead of writing appointments in an appointment book or on a calendar, engagements can be entered into the pocket computer for an audio-visual reminder. The entire list of appointments, activities, or tasks may be reviewed, edited, deleted, or printed, as well as saved on tape. Equipment needed to use the Pocket Organizer includes a PC-2 (26-3601) equipped with at least 4 KB or memory, a TRS-80 Printer/Cassette Interface (26-3605), a CCR-81 Cassette Recorder (26-1208), and blank computer cassette tapes. The Pocket Organizer costs $19.95.
Radio Shack; Ft. Worth, TX
CIRCLE NO. 210

Color Computers

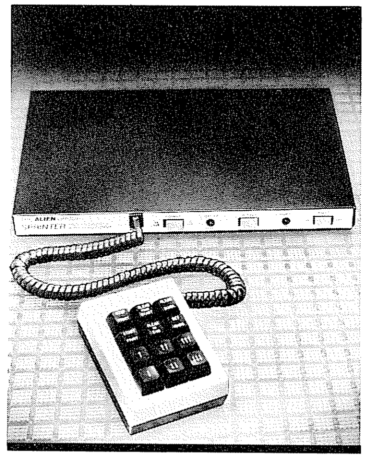
Budget management program
Cash Budget Management turns the Color Computer into a sophisticated personal or small business bookkeeping and budgeting system. By entering income and expense amounts on a regular basis using checking, petty cash, and savings accounts, Cash Budget Management makes it easy to track where money is spent for up to 12 months. The program includes up to 10 income categories, 86 expense categories, and three cash accounts, or the user may define budget categories to best fit individual needs. Information can be printed for immediate use or saved on a disk for storage purposes. A 16 KB (or greater) TRS-80 Color Computer with Extended Color Basic, a disk interface and drive, and blank diskettes are required. Cash Budget Management (26-3261) is offered for $49.95.
Radio Shack; Ft. Worth, TX
CIRCLE NO. 212

Accounting package
Mark Data Products has released a new double-entry accounting package for the Color Computer. This accounting system is for the small business-person who needs a fast, efficient means to process income and expenses, prepare detailed reports, and maintain most of the information required at tax time. The system is a family of programs that operate by means of a menu selection scheme. When the operator selects a task to perform, the computer loads a program designed to handle that task from the system disk. The system disk contains all of the programs required to create, update, and maintain data files and prepare the necessary accounting reports including a transaction journal, a P&L or income report, an interim or trial balance, and a balance sheet. Up to 255 separate accounts may be defined and a single-disk system can hold over 1,400 transactions. A machine language program is included with the system to automatically enhance the monitor screen to a 51-by-24 display. 32 KB of memory is required along with an 80-column printer and one or more disk drives. This accounting software is available on disk with a detailed operating manual for $99.95.
Mark Data Products; Mission Viejo, CA
CIRCLE NO. 213

Plotting program
Color Disk Graphics allows plotting of color charts and graphs on the Color Computer. The program's graph plotting formats include vertical or horizontal bar charts, pie charts, and line charts. Graph lines can be labeled with a key legend and charts can be saved on disk. The charts can be displayed or printed on either a low- or high-resolution Color Computer screen, any Radio Shack TRS-80 dot-matrix printer that has graphics capabilities, or a four-color graphics printer. To use Disk Graphics, a 16 KB Color Computer (26-3002) or 32 KB Color Computer (26-3003) with extended color Basic is required, along with a color television, a disk drive, and a Disk Controller Pak (26-3022). Optional equipment includes a cassette recorder for loading charts created with the ROM version of the program, a printer, and an additional disk drive (26-3023). Price: $49.95.
Radio Shack; Ft. Worth, TX
CIRCLE NO. 214

Peripherals Plus

Thermal printer
The new TP-10 Thermal Printer is ideal for use with the Micro Color Computer Model MC-10. It can also be used with the Color Computer. The whisper-quiet TP-10 prints 32 characters per line at 30 cpi on 4-1/8" wide thermal paper. It can print 95 ASCII characters plus 16 block graphics characters, compatible with screen graphics produced by the MC-10 Micro Color Computer. Features of the TP-10 include an elongation mode for expanded print and a special repeat function to make graphics programming easier. The TP-10 has a Color Computer-compatible serial interface only (600 baud). It measures 3"-by-8"-by-5" and weighs approximately 3.3 lbs. The TP-10 Thermal Printer (26-1261) is priced at $99.95 and the thermal paper (26-1332) is offered at $3.95 for a package of two rolls.
Radio Shack; Ft. Worth, TX
CIRCLE NO. 215



Print buffer
Sprinter is a unique print buffer that allows its content to be viewed on a standard monitor. Text throughout the Sprinter's 62 KB memory can be rapidly accessed. A 12-button keypad features bidirectional scrolling, as well as the ability to jump between numerous tab points set by the user. Because the Sprinter allows the user to instantly see any portion of a printout on a monitor instead of waiting for hardcopy, time and money can be saved. For example, programmers writing and debugging lengthy programs constantly pause for "latest version" printouts. In addition, they are frequently looking back and forth at various pages of the listing, referring perhaps to different branch locations. With the Sprinter, the whole program can instantly be displayed on the monitor and the programmer can rapidly view various sections using the jump controls on the keypad. The Sprinter is quite thin, and it's made of heavy-gauge steel on which a monitor can safely be placed. It comes in both serial and parallel versions and will work with any computer that drives a normal printer. No modifications whatsoever is required to the host's programs.
The Alien Group; New York, NY
CIRCLE NO. 217

Program Listings

This section is devoted to program listings from articles featured in the front portion of this issue.

Easy Menus
Continued from page 71

Listing 1

```
10 CLS
20 PRINT TAB(10) "1. DATE STONES OF RYN":PRINT
30 PRINT TAB(10) "2. TANK - AN EASY MAZE GAME":PRINT
40 PRINT TAB(10) "3. HUMAN ADVENTURE - INSIDE THE BODY":PRINT
50 PRINT TAB(10) "4. HAUNTED HOUSE - SHOOT THE GHOSTS":PRINT
60 PRINT TAB(10) "5. ASSIGNMENT 46 - A HARRY FLYNN ADVENTURE":PRINT
70 PRINT TAB(10) "6. TERMITE":PRINT
80 PRINT TAB(10) "7. 'ROUND THE END MARKER GAME":PRINT
90 PRINT TAB(10) "8. MICROTYPING";
100 A$ = INKEY$: IF A$ = "" THEN 100
110 A = VAL(A$): IF A < 1 OR A > 9 THEN 100
120 ON A GOTO 130, 140, 150, 160, 170, 180, 190, 200
130 SYSTEM"DO STONERUN"
140 RUN "TANK/BAS"
150 RUN "HUMAN/BAS"
160 RUN "HAUNT/BAS"
170 RUN "ASSGT46/BAS"
180 RUN "TERMITE/BAS"
190 RUN "ROUNDEND/BAS"
200 RUN "TYPING/BAS"
```

Listing 2

```
5 CLS
10 PRINT TAB(30) "MENU"
20 PRINT TAB(10) "1. PROFILE":PRINT
30 PRINT TAB(10) "2. TIME ACCOUNTING":PRINT
40 PRINT TAB(10) "3. SETTLEMENT AND JUDGEMENT COMPUTATION":PRINT
50 PRINT TAB(10) "4. PRINT OUT CALENDAR FOR ANY MONTH":PRINT
60 PRINT TAB(10) "5. VALUE OF AN AMOUNT, CONSIDERING TIME":PRINT
70 PRINT TAB(10) "6. TELEPHONE CRISS CROSS DIRECTORY":PRINT
80 PRINT TAB(10) "7. HORIZONTAL BAR GRAPH GENERATOR":PRINT
90 PRINT TAB(10) "8. TYPING EXERCISES";
100 CH$ = INKEY$
110 CH$ = INKEY$ : IF CH$ = "" THEN 110
120 CH = VAL(CH$)
130 ON CH GOTO 150, 160, 170, 180, 190, 200, 210, 220
140 GOTO 110
150 SYSTEM"M"
160 SYSTEM"TAS"
170 RUN "MERGED9/BAS"
180 RUN "CALENDAR/BAS"
190 RUN "TIMEINV/BAS"
200 RUN "PHONFIND/BAS"
210 RUN "MINIGRAF/BAS"
220 RUN "TYPING/BAS"
```

Personal Arcade
Continued from page 77

Subroutine 5000

```
5000 REM BUDGET, BOOKKEEPING AND ACCOUNTING ROUTINE
5010 DT#=DT#+50000:'BUDGET REVENUE (2HR PERIOD)=$50000
5020 DT#=DT#-(NP+ST)*160*PT/100:'4-MAN CREW @ $20.00/HR
5030 SC=ST*2*PT/100:'SALT CONSUMED @ 2000#/HR/TRUCK
5040 FT=FT-(NP+ST)*10*PT/100:'FUEL LEFT @ 5 GAL/HR
5050 IF TP>10 THEN SA=SA-SC ELSE CC=CC-SC:'SALT LEFT
5060 DT#=DT#-1000*25:'FIXED-COST SALARIES @ $12.50/HR
5070 DT#=DT#-PP:CR=CR-PP:IF CR<=0 THEN CR=CR
+40*(NP+ST)*(100-PT)/100: PT=100: GOTO 5100
5080 'KEEP TRACK OF REPAIRS TO TRUCKS AND PLOWS
5090 PT=PT+(100-PT)*(PP/CR):IF PT>100 THEN PT=100
5100 FT=FT+PF:DT#=DT#-1.35*PF:'FUEL PURCHASES
5110 SA=SA+PS:DT#=DT#-14*PS:'SALT+PURCHASED SALT @ $14/TON
5120 CC=CC+PC:DT#=DT#-20*PC:'CACL+PURCHASED CACL @ $20/TON
5130 DT#=DT#-1000:'RENTS, MORTGAGE, BLDG MAINTENANCE &C.
5140 IF DT#<=0 THEN GOTO 25000:'CITY IS BANKRUPT, GAME ENDS
5150 IF SA<=0 THEN SA=0:ST=0:'SALT RAN OUT. TRUCKS STOPPED
5160 IF SA<=200 THEN ?"           SALT IS RUNNING OUT"
5170 IF CC<=0 THEN CC=0:ST=0:'CACL RAN OUT. TRUCKS STOPPED
5180 IF CC<=200 THEN ?"    COLD-WEATHER SALT IS RUNNING OUT"
5190 IF FT<=0 THEN FT=0:ST=0:NP=0:'NO FUEL. NO PLOWS, SALTERS
5200 IF FT<=2000 THEN ?"     FUEL FOR TRUCKS IS RUNNING OUT"
5210 PS=0:PC=0:PF=0:PP=0:'RESET PURCHASE ORDERS
5220 RETURN
```

```
9090 Q2$=INKEY$:IF Q2$="" THEN GOTO 9090 ELSE IF Q2$="Y" OR
Q2$=" " THEN GOTO 10000
9100 GOTO 9010
9110 CLS:INPUT "HOW MANY TONS OF SALT (@ $14 PER TON)";PS
9120 RETURN
9130 CLS:INPUT "HOW MANY TONS OF CALCIUM CHLORIDE (@ $20 PER
TON)";PC
9150 RETURN
9160 CLS:INPUT "HOW MANY GALLONS OF FUEL (@ $1.35 PER
GALLON)";PF
9180 CR=CR+(NP+ST)*(100-PT)*40/100:'COST OF REPAIRS
9190 ?"TO REPAIR AND REFURBISH ALL EQUIPMENT, YOU NEED"
9200 ?"TO SPEND $";CR;"ON PARTS AND REPAIRS."
9210 IF CR<10 THEN ? "NO NEED TO SPEND ANYTHING NOW.":RETURN
9220 INPUT"HOW MUCH DO YOU WISH TO SPEND AT THIS TIME";PP
9240 RETURN
```

Subroutine 4000

```
4000 REM ROUTINE TO EMULATE WEATHER
4010 IF RND(70)=1 THEN RS=(RND(120)/100)*(RND(120)/100)
4020 TP=TP+(2-RND(3)):IF TP>32 THEN TP=32
4030 WS=RND(5)*RND(10)+RS*10
4040 IF RS=0 THEN RETURN
4050 RS=RS+0.2-Q*Q:Q=Q+0.1
4060 IF RS<0 THEN RS=0:Q=0
4070 DS=DS+RS*2
4080 DD=HD:HD=DS+(WS/10)*(WS/10):IF DD>HD THEN HD=DD
4090 RETURN
```

Subroutine 13000

```
13000 REM BUDGET REPORT
13010 CLS:?"                    2-HOUR BUDGET REPORT"
13020 ?"                   FOR";HR;A$;" ";M$;INT(DA)
13040 ?"          INCOME/ASSETS:"
13050 ?"          CASH...............$";INT(DT#)
13060 ?"          REVENUE............$ 50000"
13070 PRINT
13080 ?"          EXPENSES:"
13090 ?"          SALARIES...........$"; 1000*25
+(NP+ST)*160*PT/100
13100 ?"          SALT...............$";PS*14
13110 ?"          CALCIUM CHLORIDE...$";PC*20
13120 ?"          FUEL...............$";PF*1.35
13130 ?"          REPAIRS/PARTS......$";PP
13140 ?"          BLDG. EXPENSES.....$ 1000"
13150 PRINT
13160 ?"          DOLLARS IN TREASURY: $";INT(DT#+50000 -1000*25
-(NP+ST)*160*PT/100 -PS*14 -PC*20 -PF*1.35 -PP -1000)
13170 ?"          DO YOU WISH TO MAKE ANY PURCHASES (Y/N)?";
13180 Q1$=INKEY$:IF Q1$="" THEN GOTO 13180 ELSE IF Q1$="Y"
THEN GOTO 9000 ELSE IF Q1$=" " THEN RETURN
13190 ?:?"                  RETURN TO MENU?"
13200 Q2$=INKEY$:IF Q2$="" THEN GOTO 13200 ELSE IF Q2$="Y" OR
Q2$=" " THEN RETURN
13210 GOTO 13000
```

Subroutine 14000

```
14000 REM INVENTORY REPORT
14010 CLS:?"                    2-HOUR INVENTORY REPORT"
14020 ?"                   FOR";HR;A$;" ";M$;INT(DA)
14030 PRINT
14040 ?"          SALT................";SA;"TONS"
14050 ?"          CALCIUM CHLORIDE....";CC;"TONS"
14060 ?"          FUEL................";FT;"GALLONS"
14070 ?"          PLOWS WORKING.......";INT(NP*PT/100)
14080 ?"          SALT TRUCKS WORKING.";INT(ST*PT/100)
14090 ?"          SPARE PARTS, ETC...$";PP-CR;"WORTH"
14100 ?"          DO YOU WISH TO MAKE ANY PURCHASES (Y/N)?"
14110 Q1$=INKEY$:IF Q1$="" THEN GOTO 14110 ELSE IF Q1$="Y"
THEN GOTO 9000 ELSE IF Q1$=" " THEN RETURN
14120 ?"                  RETURN TO MENU?"
14130 Q2$=INKEY$:IF Q2$="" THEN GOTO 14130 ELSE IF Q2$="Y" OR
Q2$=" " THEN RETURN
14140 GOTO 14000
```

Subroutine 9000

```
9000 REM REPLENISHMENT ROUTINE (PURCHASING)
9010 CLS:?"ITEM TO PURCHASE:"
9020 ?"1. SALT"
9030 ?"2. CALCIUM CHLORIDE"
9040 ?"3. TRUCK FUEL/REPAIR PARTS
9050 I$=INKEY$:IF I$="" THEN GOTO 9050
9060 R=VAL(I$):IF R >3 OR R<1 THEN GOTO 9010
9070 ON R GOSUB 9110,9130,9160
9080 ?"RETURN TO MENU?";
```

Subroutine 11000

```
11000 CLS:?:?"          WEATHER REPORT FOR";"     ";HR;A$;"
";M$;INT(DA)
11010 ?"          RATE OF SNOWFALL";RS;"INCHES PER HOUR"
11020 ?"          WIND SPEED";WS;"MILES PER HOUR"
11030 ?"          TEMPERATURE";TP;"DEGREES FAHRENHEIT"
11040 ?"          TOTAL ACCUMULATED SNOWFALL";DS;"INCHES"
11050 ?"          DRIFTS TO";INT(0.5+HD/12);"FEET"
11060 GOTO 19040
```

Subroutine 6000

```
6000 REM "STREET CONDITION" ROUTINE
6010 PR=PR -(HD*WS*.002+RS*WS)/5 +((NP+ST)*PT/100)/50
6020 IF PR<0 THEN PR=0 ELSE IF PR>100 THEN PR=100
6030 'LINES ABOVE CALCULATE PERCENT OF ROADS OPEN
6040 IF TP<0 THEN PT=PT+2*TP
6050 IF PT<0 THEN PT=0 ELSE IF PT>100 THEN PT=100
6060 IF RND(5)=2 THEN PT=PT-2*(RND(3))
6070 'PERCENTAGE OF TRUCKS FUNCTIONING
6080 CR=CR +40*(NP+ST)*(100-PT)/100: 'COST OF REPAIRS
6090 RETURN
```

Subroutine 12000

```
12000 'STREET REPORT
12010 CLS:?"                    2-HOUR STREET REPORT"
12020 ?"                   FOR";HR;A$;" ";M$;INT(DA)
12030 ?:?"   ";PR;"% OF CITY STREETS ARE OPEN"
12040 ?"   ";INT(NP*PT/100);"SNOWPLOWS (OR GARBAGE TRUCKS)"
12050 ?"    ARE CLEARING ROADS (OR COLLECTING GARBAGE)"
12060 ?"   ";INT(ST*PT/100);"SALT TRUCKS ARE SALTING STREETS"
12070 ?"   "; PR*PT/100;"% OF STREETS ARE FREE OF ICE"
12080 ?"   ";G;"% OF THE GARBAGE HAS BEEN PICKED UP"
12090 IF DG<2 THEN GOTO 19040
12100 ?"   GARBAGE HAS BEEN PILING UP FOR";INT(DG+.5);"DAYS"
12110 GOTO 19040
```

Subroutine 15000 and 16000

```
15000 REM CALLOUT 50 PLOWS
15010 NP=50:GOSUB 8000
15020 ?"                  50 PLOWS CALLED OUT"
15030 ? "             ";PR;"% OF CITY STREETS ARE OPEN"
15040 GOTO 19040

16000 REM CALLOUT 100 PLOWS
16010 NP=100:GOSUB 8000
16020 ?"     100 PLOWS CALLED OUT, INCLUDING GARBAGE TRUCKS"
16030 ? "             ";PR;"% OF CITY STREETS ARE OPEN"
16040 GOTO 19040
```

Subroutine 17000

```
17000 REM CALLOUT 50 SALT TRUCKS
17010 ST=50:GOSUB 8000
17020 ?"               50 SALT TRUCKS CALLED OUT"
17030 ?"          ";PR*PT/100;"% OF STREETS ARE ICE-FREE"
17040 GOTO 19040
```

Subroutine 18000

```
18000 REM CALLOUT 100 SALT TRUCKS
18010 ST=100:GOSUB 8000
18020 ?"                100 SALT TRUCKS CALLED OUT"
18030 ?"              ";FR*PT/100;"%  OF STREETS ARE ICE-FREE"
18040 GOTO 19040
```

Subroutine 19000

```
19000 REM CALL FOR GARBAGE PICKUP
19010 G=G+10*(PR/100)*(PT/100):IF G>101 THEN NP=0: G=101: ELSE
NP=50*(PT/100)
19020 GOSUB 8000: ST=0
19030 ?"            GARBAGE PICKUP ";G;"% COMPLETE"
19040 ?"                PRESS SPACEBAR TO RETURN"
19050 I$=INKEY$:IF I$="" THEN GOTO 19050 ELSE RETURN
```

Subroutine 8000

```
8000 REM "ADD TWO HOURS" TIMEKEEPING ROUTINE
8010 HR=HR+2:IF HR=14 THEN HR=2:DA=DA+0.5:T$=A$:A$=P$:P$=T$
8020 IF DA>31.5 AND M$="January" THEN DA=1:M$="February"
8030 IF DA>28.5 AND M$="February" THEN DA=1:M$="March"
8040 IF DA>27.5 AND M$="March" THEN GOTO 25000:'END ROUTINE
8050 CLS:PRINT@ 12,HR;A$;"   ";M$;INT(DA)
8060 IF PL=3 THEN N=N+.25: IF INT(N)=4 THEN N=1
8070 IF PL=2 THEN N=N+.16666:IF INT(N)=3 THEN N=1
8080 ?"           COMMISSIONER: ";P$(INT(N))
8090 G=G-2: GOSUB 5000: GOSUB 4000: GOSUB 6000
8100 IF G<10 THEN GOSUB 21000:'GARBAGE PILEUP ALERT
8110 IF RS>0.3 AND Q=0.1 THEN GOSUB 22000:'STORM ALERT
8120 RETURN
```

Subroutine 21000

```
21000  REM GARBAGE PICKUP OVERDUE--MESSAGE
21010 ? "           GARBAGE IS PILING UP! SEND TRUCKS!"
21020 DG=DG+.08
21030 RETURN
```

Subroutine 22000

```
22000 REM STORM ALERT WEATHER BULLETIN
22010 ?"              ***** STORM ALERT *****"
22020 ?"           THE NATIONAL WEATHER BUREAU HAS"
22030 ?"           JUST ISSUED A BLIZZARD WARNING"
22040 ?"           FOR YOUR METROPOLITAN AREA."
22050 ?"           UP TO";30*RS;"INCHES OF SNOW WITH"
22060 ?"           GALE FORCE WINDS UP TO";WS+25;"MILES"
22070 ?"           PER HOUR ARE EXPECTED DURING THE"
22080 ?"           NEXT 24 HOUR PERIOD.  THE STORM"
22090 ?"           IS EXPECTED TO ARRIVE WITHIN"
22100 ?"           THE NEXT TWO HOURS."
22110 RETURN
```

Subroutine 20000

```
20000 REM SNOOZE ROUTINE
20010 NP=0:ST=0:GOSUB 8000:'CALL OFF TRUCK CREWS
20020 IF RS>.6 THEN ?"           HOW CAN YOU SLEEP AT A TIME
LIKE THIS?" ELSE ?"                 HAVE A NICE NAP"
20030 ?"            PRESS SPACEBAR TO WAKE UP"
20050 GOTO 19050
```

Polynomial Functions: A Different View
Continued from page 85

Listing 1. *PC-3 version*

IMPORTANT NOTE
Do not enter REM statements.
Line length or memory will be exceeded.
REM's are included only to make the program easier to understand.

Always use NEW before loading the program.
Use PRINT = LPRINT, if output is to be directed to the printer.
At the prompt, OPT?, enter the letter of the segment required or use DEF N as needed, where N is the letter of the option required.

```
405:INPUT "START AT X=? ";X:REM Get starting
    point for X.  Function must be positive
    at this point.
   :GOSUB 500:REM Test value of function
    at this point.  If negative, routine
    will fail.
   :IF Y>0 THEN 415
410:PRINT "Y MUST BE >0 RETRY"
   :GOTO 405
415:INPUT "INCREMENT FOR X? ";V:REM Get
    increment for X.  Routine expects plus
    or minus one, if loop is to count
    correctly.
   :FOR Z=1 TO U+1:REM Loop for required
    number of digits.
420:X=X+V:REM Get next X.
   :GOSUB 500:REM Evaluate function.
   :IF Y>0 THEN 420:REM If function is
```

```
*** MENU ***
  5:"M":PRINT "SOLVE IT"
    :PRINT "OPTIONS ARE-"
    :PRINT "L-LOAD COEFFICIENTS"
 10:PRINT "D-DISPLAY COEFFICIENTS"
    :PRINT "V-VALUES OF FUNCTION"
    :PRINT "S-SOLVE FOR ROOT"
 15:INPUT "OPT? ";Z$
    :GOTO Z$


*** LOAD COEFFICIENTS ***
100:"L":PRINT "LOAD COEFFICIENTS"
    :INPUT "DEGREE? ";W:REM Get degree of
     function.
    :FOR Z=1 TO W+1:REM Get coefficients.
105:INPUT "COEFF? ";A(Z)
    :NEXT Z:REM Continue to Display
     Coefficients.

*** DISPLAY COEFFICIENTS ***
200:"D":PRINT "DISPLAY COEFFICIENTS"
    :FOR Z=1 TO W+1:REM Loop for number of
     coefficients plus constant term.
    :PRINT Z;" ";A(Z)
    :NEXT Z
    :GOTO 5:REM Return to menu.

*** VALUES OF FUNCTION ***
300:"V":PRINT "VALUES OF FUNCTION"
    :PRINT "BREAK FOR EXIT"
305:INPUT "INCREMENT FOR X? ";V:REM Get
     increment for X, commonly plus or
     minus one.
    :INPUT "START AT X=? ";X:REM Get initial
     value, usually zero.
310:GOSUB 500:REM Evaluation begins here.
     BREAK for exit.
    :PRINT "X=";X;" Y=";Y:REM Display
     computed value.
    :X=X+V:REM Increment X.
    :GOTO 310:REM Return for next computation.

*** SOLVE FOR ROOT ***


400:"S":PRINT "SOLVE FOR ROOT"
    :INPUT "ACCURACY REQUIRED? ";U:REM Get
     accuracy required, number of decimal
     positions.
```

```
     positive, X has not passed root.
425:X=X-V:REM Here if root passed; function
     now negative.  Get previous value of X.
    :PRINT "X=";X:REM Display current
     approximation.
    :V=V/10:REM Decrease increment by a
     factor of ten.
    :NEXT Z:REM Return for next
     approximation.
    :GOTO 405:REM SInce frequently more than
     one root is required, this segment
     returns to itself, instead of the menu.
     Interrupt with BREAK and/or option
     selection with DEF.

*** SUB: EVALUATE FUNCTION ***
500:Y=A
    :FOR T=2 TO W+1
    :Y=Y*X+A(T)
    :NEXT T
    :RETURN

(END PC-3 LISTING: SOLVE IT)
```

Listing 2. *PC-4 version*

IMPORTANT NOTE
Do not enter REM statements.
Line length or memory will be exceeded.
REM's are included only to make the program easier to understand.

Use CLEAR before loading, in the program area to be used.
Use Mode 7, if output is to be directed to the printer.

The program can be loaded into separate program areas for easy access with SHIFT N, where N is the number of the program area required. Note the subroutine at line 500 is required to generate ordered pairs (VALUES) and to solve for roots (SOLVE). Thus the subroutine will be required in each segment if loaded into separate program areas.

```
***        MENU        ***
*** PROGRAM AREA PO ***
  5 PRINT "SOLVE IT"
   :PRINT "OPTS ARE-"
   :PRINT "1-LOAD":REM Load coefficients
```

```
 10 PRINT "2-DISPLAY":REM Display
    coefficients loaded.
   :PRINT "3-VALUES":REM Compute values
    of function for plot.
   :PRINT "SOLVE":REM Find roots.
 15 INPUT "OPT #",Z
   :GOTO 100*Z

*** LOAD COEFFICIENTS ***
100 PRINT "LOAD"
   :INPUT "DEGREE",W:REM Get degree of
    function.
   :FOR Z=1 TO W+1:REM Get coefficients
105 INPUT "COEFF",A(Z)
   :NEXT Z:REM Continue to display
    coefficients.

*** DISPLAY COEFFICIENTS ***
200 PRINT "DISPLAY"
   :FOR Z=1 TO W+1:REM Loop for number of
    coefficients.
   :PRINT Z;" ";A(Z)
   :NEXT Z
   :GOTO 5:REM Return to menu.

*** VALUES OF FUNCTION ***
300 PRINT "VALUES"
   :PRINT "USE STOP"
305 INPUT "INCREMENT",V:REM Get increment
    for X, commonly plus or minus one.
   :INPUT "START AT X=",X:REM Get initial
    value, usually zero.
310 GOSUB 500:REM Evaluation begins here.
    Use STOP for exit.
   :PRINT "X=";X:REM Display computed values.
   :PRINT "Y=";Y
   :X=X+V:REM Increment X.
   :GOTO 310:REM Return for next
    computation.

*** SOLVE FOR ROOT ***

400 PRINT "SOLVE"
   :INPUT "ACCURACY",U:REM Get accuracy
    required, number of decimal positions.
405 INPUT "START AT X=",X:REM Get starting
    point for X.  Function must be positive
    at this point.
   :GOSUB 500:REM Test value of function
    at this point.  If negative, routine
    will fail.
   :IF Y>0 THEN 415
410 PRINT "Y<0 RETRY"
   :GOTO 405
415 INPUT "INCREMENT",V:REM Get increment
    for X.  Routine expects plus or minus
    one, if loop is to count correctly.
   :FOR Z=1 TO U+1:REM Loop for required
    number of digits.
420 X=X+V:REM Get next X.
   :GOSUB 500:REM Evaluate function.
   :IF Y>0 THEN 420:REM If function is
    positive, X has not passed root.
425 X=X-V:REM Here if root passed; function
    now negative.  Get previous value of X.
   :PRINT "X=";X:REM Display current
    approximation.
   :V=V/10:REM Decrease increment by a
    factor of ten.
   :NEXT Z:REM Return for next
    approximation.
   :GOTO 405:REM Since frequently more than
    one root is required, this segment
    returns to itself, instead of the menu.
    Interrupt with STOP.

*** SUB: EVALUATE FUNCTION ***
500 Y=A(1)
   :FOR T=2 TO W+1
   :Y=Y*X+A(T)
   :NEXT T
   :RETURN

(END PC-4 LISTING: SOLVE IT)
```

Variable Names

A – S: Available for coefficients of polynomial function
T = Temporary value
U = Accuracy required for root as number of decimal positions
V = Increment for X
W = Degree of polynomial function. Y = F(X)
Y = Range of Y = F(X)
Z = Temporary value

65024: Peek the Magic Number
Continued from page 89

Listing 1

```
100 REM * Screen Peek Demo * Model 100/8k * Richard Ramella
110 CLEAR 100
120 CLS
130 PRINT @ 1,"APE";
140 PRINT
150 INPUT "Which letter do we check - 1, 2 or 3";X
160 IF X=1 OR X=2 OR X=3 THEN 200
170 PRINT @ 40,STRING$(90,224);
180 PRINT @ 40,"";
190 GOTO 150
200 PRINT
210 Z=PEEK(65024+X)
220 PRINT "ASCII value:"Z" Letter: "CHR$(Z)
230 PRINT "Peek position:"65024+X
240 END
```

Listing 2

```
100 REM * Hubert Finds the Ball * TRS-80 Model 100/8K * Ramella *
110 CLEAR 300
120 CLS
130 PK=65024
140 H$(1)=SPACE$(1)+CHR$(147)
150 H$(2)=SPACE$(1)+CHR$(148)
160 PRINT @ 0,H$(1)
170 PRINT "I'm Hubert. Hide the ball from me
180 PRINT "anywhere from 1 to 279."
190 INPUT "Where";X
200 IF X<1 OR X>279 THEN PRINT @ 80,STRING$(90,224);:
    PRINT @ 80,"Please... ";: GOTO 180
210 PRINT @ 40,STRING$(255,224);
220 FOR Z=1 TO 20
230 PRINT @ RND(1)*269+10,"x";
240 NEXT Z
250 PRINT @ X,"O";
260 FOR J=1 TO 278
270 PRINT @ J,H$(1);
280 FOR T=1 TO 20
290 NEXT T
300 Q=PEEK(PK+J+2)
310 IF Q=120 THEN GOSUB 340
320 IF Q=79 THEN 440
330 NEXT J
340 PRINT @ 280,"That's not a ball!";
350 FOR F=1 TO 5
360 FOR T=1 TO 2
370 PRINT @ J,H$(T);
380 SOUND 1000+RND(1)*1000,2
390 FOR N=1 TO 50
400 NEXT N
410 NEXT T,F
420 PRINT @ 280,STRING$(30,224);
430 RETURN
440 PRINT @ 280,"I found it!";
450 FOR T=1 TO 2
460 PRINT @ J,H$(T);
470 SOUND 1000+RND(1)*10*10,1
480 NEXT T
490 GOTO 450
500 END
```

Listing 3

```
100 REM * Cloud Rider * TRS-80 Model 100 8K * Richard Ramella *
110 CLEAR 2000
120 DEFSTR A-C
130 FOR S=1 TO VAL(RIGHT$(TIME$,2))
140 T=RND(1)
150 NEXT
160 Q=65024
170 A=CHR$(133)
180 DIM B(75)
190 FOR Z=1 TO 75
200 B(Z)=STRING$(25,224)
210 NEXT
220 FOR X=0 TO 280 STEP 40
230 PRINT @ X,STRING$(39,239);
240 NEXT X
250 PRINT @ 52," Cloud    "
260 PRINT @ 92," Rider    "
270 PRINT @ 132," Must rev "
280 PRINT @ 172," 1 minute."
290 PRINT @ 212," Patience "
300 GOSUB 590
310 FOR L=10 TO 290 STEP 40
320 PRINT @ L,STRING$(25,224);
330 NEXT
340 W=292
350 PRINT @ 289,"*";
360 Z=67
370 K=10
380 FOR X=Z TO Z+7
390 PRINT @ K,B(X);
```

65024: Peek the Magic Number

Listing 3 (continued)

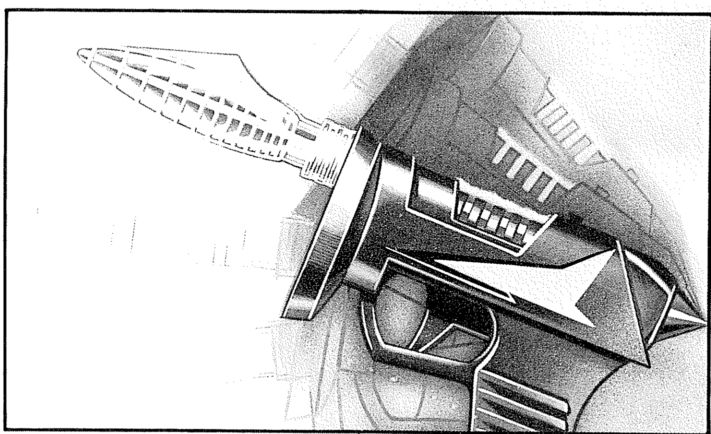
```
400 K=K+40
410 C=INKEY$
420 S=W
430 IF W>290 AND C="," THEN W=W-1: PRINT @ S,CHR$(224);
440 IF W<314 AND C="." THEN W=W+1: PRINT @ S,CHR$(224);
450 PRINT @ W,A;
460 IF PEEK(Q+W-1)=42 OR PEEK(Q+W+1)=42 THEN GOSUB 730
470 IF PEEK(Q+W-40)<>224 THEN GOSUB 660
480 NEXT X
490 D=D+1
500 PRINT @ 280,D;
510 Z=Z-1
520 IF Z>0 THEN 370
530 PRINT @ 12,"Storm intensifies";
540 FOR G=1 TO 68
550 N=INT(RND(1)*2)
560 IF N=1 THEN MID$(B(G),1+RND(1)*22,3)=STRING$(RND(1)*4,255)
570 NEXT G
580 GOTO 360
590 FOR Z=1 TO 67
600 FOR X=1 TO 25
610 J=INT(RND(1)*Z/2)
620 IF J=1 AND Z<67 THEN MID$(B(Z),X,1)=CHR$(251+(RND(1)*4))
630 NEXT X
640 NEXT Z
650 RETURN
660 PRINT @ 280,STRING$(9,239);
670 PRINT @ 160,"Final";CHR$(32);
680 PRINT @ 200,"Score:";
690 SOUND RND(1)*1000+1200,1
700 PRINT @ W,CHR$(224+RND(1)*15);
710 RT=RT+1
720 IF RT<30 THEN 690 ELSE END
730 SS=SS+(D*2)
740 PRINT @ 240,SS
750 FOR T=1300 TO 1000 STEP -25
760 SOUND T,1
770 NEXT
780 IF PEEK(Q+289)<>42 THEN 820
790 PRINT @ 289,CHR$(239);
800 PRINT @ 315,"*";
810 RETURN
820 PRINT @ 315,CHR$(239);
830 PRINT @ 289,"*";
840 RETURN
850 END
```

Biorythms and Your CoCo

Continued from page 97

```
1 REM BIORHYTHM BY ROBERT TOSCANI,4744 WHITAKER AVE. PHILA. PA. 19120
10 G=0:K=0:L=0:U=0:J=0:H=0:V=0:CLS
15 INPUT"GRAPHICS OR NUMBERS G/N";F$
20 IFF$<>"G"ANDF$<>"N"THEN15
25 INPUT"YEAR BORN";A
26 INPUT"MONTH BORN";B
27 INPUT"DAY BORN";C
28 INPUT"CHART YEAR";D
29 INPUT"CHART MONTH";E
30 IF B=E THEN 80
35 IF B<E THEN U=B:GOTO65
40 FOR T=B TO 12
45 ON B GOSUB235,240,235,245,235,245,235,235,245,235,245,235
50 G=G+J:NEXT
55 U=1
60 IFE=1THEN80
65 FOR I=U TO E-1
70 ON I GOSUB 235,240,235,245,235,245,235,235,245,235,245,235
75 L=L+J:NEXT
80 H=G-C:F=D-A
85 IF B>E THEN F=F-1
90 F=INT(F*365.25)+H+L
95 M=F-(INT(F/23)*23):N=F-(INT(F/28)*28):O=F-(INT(F/33)*33)
100 IFE=13THENE=1:D=D+1
105 CLS:R=-1:IFF$="G"THEN115ELSEGOSUB230
110 R=0
115 ON E GOSUB235,240,235,245,235,245,235,235,245,235,245,235
120 IF J=28THENINPUT"IS THIS A LEAP YEAR Y/N";G$:IFG$="Y"THENJ=29
125 IF F$="G"THEN250
130 FORP=1TOJ
135 IFM=23THENM=0
140 IFN=28THENN=0
145 IFO=33THENO=0
150 M=M+1:N=N+1:O=O+1
155 IF F$="G"THENGOSUB285:NEXT:GOTO215
160 IFM<=11THENM$="+"ELSEM$="-"
165 IFM=11ORM=23THENR=R+1:PRINT@68+R*32,M$;:PRINT@68+(1+R)*32,P;
170 IFN<=14THENN$="+"ELSEN$="-"
175 IFN=14ORN=28THENR=R+1:PRINT@77+R*32,N$;:PRINT@77+(1+R)*32,P;
180 IFO<=16THENO$="+"ELSEO$="-"
185 IFO=16ORO=33THENR=R+1:PRINT@86+R*32,O$;:PRINT@86+(1+R)*32,P;
190 NEXT
195 PRINT@417,"CORRECTION P/M";:INPUTB$
200 IFB$="P"THENV=V-1ELSEIFB$="M"THENV=V+1ELSEGOTO215
210 M=M+V:N=N+V:O=O+V:E=E+1:GOTO100
215 PRINT@0,"NEXT Y/N";:INPUTA$:IF A$="Y"THENE=E+1:GOTO100
220 PRINT@483,"ANYBODY ELSE Y/N";:INPUT C$:IF C$="Y"THEN 10
225 END
230 PRINT@14,E;"/";D:PRINT@36,"PHY      EMOT     INT":PRINT@29,V;:RETURN
235 J=31:RETURN
240 J=28:RETURN
245 J=30:RETURN
250 CLS0:FORZ=1TO63:SET(Z,15,5):NEXT
255 PRINT@14,E;"/";D;:PRINT@32,"+";:PRINT@448,"-";
260 PRINT@417,"123456789";:PRINT@427,"123456789";:PRINT@437,"123456789";
265 PRINT@426,"0";:PRINT@436,"0";:PRINT@458,"1";:PRINT@468,"2";:IFJ<30THEN130
275 PRINT@446,"0";:PRINT@478,"3";:IFJ<31THEN130ELSEPRINT@447,"1";:GOTO130
285 IFM<=6THENAM=-M ELSEIFM=>6ANDM<=11THENAM=M-11ELSEIFM>11ANDM<=17THENAM=M-10ELSEIFM>17THENAM=23-M
290 IFN<=7THENAN=-N ELSEIFN>7ANDN<=14THENAN=N-14ELSEIFN>14ANDN<=21THENAN=N-13ELSEIFN>21THENAN=28-N
295 IFO<=8THENAO=-O ELSEIFO=>8ANDO<=16THENAO=O-16ELSEIFO>16ANDO<=24THENAO=O-15ELSEIFO>24THENAO=33-O
300 SET(P*2,15+AM,4):SET(P*2,15+AN,3):SET(P*2,15+AO,2)
305 RETURN
```

A Disk Zapper for the CoCo
Continued from page 93



```
*************************
*       CCZAP  V1.1     *
* (C)1983 BY J. L. POST *
*************************
*
*** SYSTEM EQUATES ***
CLS        EQU      $A928
CHROUT     EQU      $A282
CHRIN      EQU      $A1B1
KBSCAN     EQU      $A1C1
DLY        EQU      $A7D3
VIDEO      EQU      $400
DSKCON     EQU      $C004
DCB        EQU      $C006
*
**************************
* PROGRAM INITIALIZATION *
**************************

*
           ORG      $1000
START      JSR      CLS
           LDU      #DCB        U POINTS TO DSKCON DATA
           LEAX     MSG,PCR     X => INIT MESSAGE
           LBSR     PRINT       PRINT ASCII
           LDA      SECTR,PCR   PRINT CURRENT SECTOR
           LBSR     PHEX        HEX PRINT ROUTINE
           LDX      $88         CURSOR LOCATION
           LEAX     -34,X       CHANGE TO WHERE
           STX      $88         TRACK IS TO BE PRINTED
           LDA      TRACK,PCR   DO THE SAME FOR THE
           LBSR     PHEX        CURRENT TRACK #
           LDX      $88
           LEAX     -34,X
           STX      $88
           LDA      DRIVE,PCR   AND FOR THE CURRENT
           LBSR     PHEX        DRIVE #
           LDX      $88
           LEAX     -34,X
           STX      $88
           TST      ERFLG,PCR   SEE IF ERROR ON LAST ACCESS
           BEQ      MAINLP      GO IF NOT
           TFR      X,U         ELSE SAVE CURSOR LOCATION
           LEAX     -40,X
           STX      $88
           LEAX     ERMSG,PCR   AND PRINT I/O ERROR MESSAGE
           LBSR     PRINT
           STU      $88         RESTORE CURSOR LOCATION
           CLR      ERFLG,PCR   CLEAR ERROR AND CONTINUE
*
********************************
* MAIN CONTROL LOOP - DETERMINE *
* FUNCTION AND BRANCH TO ROUTINE*
********************************
*
MAINLP     LBSR     GETCHR      GET INPUT FROM KEYBOARD
           CMPA     #82         IS IT 'R'?
           BEQ      READ        GO READ SECTOR
           CMPA     #87         IS IT 'W'?
           BEQ      WRITE       GO WRITE SECTOR
           CMPA     #77         IS IT 'M'?
           LBEQ     MODIFY      GO MODIFY BUFFER
           CMPA     #59         IS IT ';'?
           LBEQ     INCRSC      GO INCREMENT SECTOR
           CMPA     #45         IS IT '-'?
           LBEQ     DECRSC      GO DECREMENT SECTOR
           CMPA     #66         IS IT 'B'?
           BNE      MAINLP      REDO IF NOT; ELSE
           RTS                  RETURN TO BASIC
*
***********************
* READ SECTOR ROUTINE *
***********************
*
```

```
READ     BSR     GETDAT      GET DSKCON DATA
         LDA     #2          DSKCON READ FUNCTION CODE
         BRA     DSKINT
*
************************
* WRITE SECTOR ROUTINE *
************************
*
WRITE    BSR     GETDAT      GET DSKCON DATA
         LDA     #3          DSKCON WRITE FUNC CODE
DSKINT   STA     0,X         STORE FUNCTION CODE
         LDA     DRIVE,PCR
         STA     1,X         STORE DRIVE #
         LDD     TRACK,PCR
         STD     2,X         STORE TRACK AND SECTOR
         LEAU    BFR,PCR     POINT TO INPUT BUFFER
         STU     4,X         TELL DSKCON WHERE IT IS
         JSR     [DSKCON]    EXECUTE FUNCTION
         TST     6,X         CHECK FOR ERRORS
         LBEQ    START       GO IF NO ERROR
         COM     ERFLG,PCR   FLAG ERROR
         LBRA    START       AND GO REPORT IT
*
*************************************
* GET PARAMETERS FOR DSKCON ROUTINE *
*************************************
*
GETDAT   JSR     CHROUT      ECHO 'R' OR 'W' TO SCREEN
         LDX     $88         GET CURSOR LOCATION
         LEAX    31,X        POINT TO DRIVE # AREA
         STX     $88
         TFR     X,U         SAVE CURSOR LOCATION
         LBSR    HEXIN       GET HEX NUMBER INTO X REG
         BCS     DATBRK      GO IF BREAK KEY HIT
         TSTB                ANYTHING ENTERED?
         BEQ     SKIP1       NO CHANGE IF NO ENTRY
         TFR     X,D
         STB     DRIVE,PCR   SAVE DRIVE #
         STU     $88         RESTORE CURSOR LOCATION
         TFR     B,A
         LBSR    PHEX        PRINT NEW DRIVE #
SKIP1    LEAU    32,U        BUMP CURSOR TO
         STU     $88         NEXT LINE
         LBSR    HEXIN       GET TRACK # IN HEX
         BCS     DATBRK      GO IF BREAK KEY HIT
         TSTB                ANYTHING ENTERED?
         BEQ     SKIP2       GO IF NO CHANGE
         TFR     X,D
         STB     TRACK,PCR   SAVE NEW TRACK #
         STU     $88         RESTORE CURSOR
         TFR     B,A
         LBSR    PHEX        PRINT NEW TRACK #
SKIP2    LEAU    32,U        BUMP CURSOR TO
         STU     $88         NEXT LINE
         LBSR    HEXIN       GET SECTOR # IN HEX
         BCS     DATBRK      GO IT BREAK KEY HIT
         TSTB                ANYTHING ENTERED?
         BEQ     SKIP3       GO IF NOT
         TFR     X,D
         STB     SECTR,PCR   SAVE NEW SECTOR #
         STU     $88         RESTORE CURSOR LOCATION
         TFR     B,A
         LBSR    PHEX        PRINT NEW SECTOR #
SKIP3    LDX     DCB         X => DSKCON DCB AREA
         RTS                 RETURN TO CALLER
DATBRK   LEAS    2,S         CLEAN UP STACK
         LBRA    START       AND RESTART
*
************************
* MODIFY BUFFER ROUTINE *
************************
*
MODIFY   LDX     #$400       POINT TO START OF VIDEO MEM
         STX     $88         CURSOR TO START
         LBSR    SCRASC      PUT ASCII ON SCREEN
MODLP    LBSR    GETKEY      GET CHAR FROM KEYBOARD
         CMPA    #3          IS IT BREAK?
         LBEQ    START       RESTART IF SO
         CMPA    #12         IS IT CLEAR?
         LBEQ    SWAP        SWAP SCREENS IF SO
         CMPA    #92         IS IT SHIFT CLEAR?
         LBEQ    MODDMP      DUMP TO PRINTER IF SO
         LBSR    CURS        GO SEE IF ARROW KEY
         BCC     MODLP       REDO IF ARROW KEY
         TST     MODE,PCR    ELSE MODIFY DATA
         BEQ     MODHEX      GO IF IN HEX MODE
         PSHS    X,D         ELSE SAVE REGS
         LBSR    COMPAD      GET ADRS OF CHAR INTO X
         PULS    D           RECOVER CHARACTER
         STA     0,X         PUT INTO BUFFER
         PULS    X           RECOVER X REG
         LBSR    SCRASC      UPDATE SCREEN
MOVER    LDA     #9
         LBSR    CURS        UPDATE CURSOR LOCATION
         BRA     MODLP       GO GET NEXT CHAR
MODHEX   CMPA    #$30        CHECK FOR VALID HEX CHAR
         BLO     MODLP
         CMPA    #$3A
         BLO     MDHX1
         CMPA    #$41
         BLO     MODLP
         CMPA    #$47
         BHS     MODLP
         SUBA    #7
MDHX1    ANDA    #$0F        CONVERT FIRST CHARACTER
         LSLA                TO BINARY DIGIT
         LSLA                AND SHIFT INTO
         LSLA                HIGH NIBBLE
         LSLA
         PSHS    A,X         SAVE IT
         LBSR    COMPAD      GET CHAR ADRS INTO X REG
         LDB     0,X         GET CURRENT HEX DATA
```

```
01860           ANDB    #$0F        MASK LOW NIBBLE
01870           ORB     0,S         ADD NEW HIGH NIBBLE
01880           STB     0,X         PUT INTO BUFFER
01890           LBSR    SCRHEX      UPDATE SCREEN
01900           LDA     #9
01910           LBSR    CURS        UPDATE CURSOR
01920 MDHX2     LBSR    GETKEY      GET NEXT NIBBLE FROM KYBRD
01930           CMPA    #$30        CHECK FOR VALID HEX #
01940           BLO     MDHX2       REPEAT IF NOT HEX
01950           CMPA    #$3A
01960           BLO     MDHX3
01970           CMPA    #$41
01980           BLO     MDHX2
01990           CMPA    #$47
02000           BHS     MDHX2
02010           SUBA    #7
02020 MDHX3     ANDA    #$0F        MASK OFF HIGH NIBBLE
02030           ORA     0,S         ADD TO PREVIOUS NIBBLE
02040           STA     0,S         SAVE IT
02050           LBSR    COMPAD      GET BUFFER ADRS INTO X
02060           PULS    A           RECOVER HEX DATA
02070           STA     0,X         SAVE HEX DATA IN BUFFER
02080           LBSR    SCRHEX      UPDATE SCREEN
02090           LDA     #9
02100           LBSR    CURS        UPDATE CURSOR
02110           LBSR    CURS
02120           PULS    X           RECOVER X REG
02130           LBRA    MODLP       GO GET MORE HEX
02140 *
02150 ***********************************
02160 * CHANGE SCREEN FROM HEX TO ASCII *
02170 *       DISPLAY OR VICE VERSA     *
02180 ***********************************
02190 *
02200 SWAP      LDA     MODE,PCR    FIND OUT WHICH MODE
02210           BEQ     SWASC       GO IF HEX TO ASCII
02220           LBSR    SCRHEX      MAKE SCREEN HEX DISPLAY
02230           LBRA    MODLP       AND RETURN TO MODIFY RTN
02240 SWASC     LBSR    SCRASC      MAKE SCREEN ASCII DISPLAY
02250           LBRA    MODLP       AND RETURN TO MOD RTN
02260 *
02270 **********************************
02280 * DUMP SCREEN TO PRINTER ROUTINE *
02290 **********************************
02300 *
02310 MODDMP    LDA     #$FE        PRINTER CODE (-2)
02320           STA     $6F         INTO DEVICE FLAG
02330           LEAX    LPTRK,PCR   GET TRACK MESSAGE
02340           LBSR    PRINT       SEND TO PRINTER
02350           LDA     TRACK,PCR   GET TRACK #
02360           LBSR    PHEX        AND PRINT IT
02370           LEAX    LPSEC,PCR   DO AGAIN FOR SECTOR #
02380           LBSR    PRINT
02390           LDA     SECTR,PCR
02400           LBSR    PHEX
02410           LDA     #13
02420           JSR     CHROUT      PRINT A CARRIAGE RETURN
02430           LDA     #10
02440           JSR     CHROUT      AND A LINE FEED
02450           LEAX    BFR,PCR     POINT TO BUFFER
02460           CLRB                SET UP COUNT FOR 256 BYTES
02470 DASCLP    LDA     ,X+         GET BYTE FROM BUFFER
02480           TST     MODE,PCR    WHICH MODE?
02490           BEQ     DMPHEX      GO IF HEX MODE
02500           ANDA    #$7F        CONVERT GRAPHICS CODES
02510           CMPA    #32
02520           BHS     DALP1       GO IF PRINTABLE CODE
02530           LDA     #46         ELSE MAKE IT A PERIOD (.)
02540 DALP1     JSR     CHROUT      AND PRINT IT
02550 DALP2     LDA     #32
02560           JSR     CHROUT      PRINT A SPACE
02570           INCB                INCREMENT COUNTER
02580           BEQ     DADONE      GO IF 256 BYTES DONE
02590           BITB    #$0F        16 BYTES DONE?
02600           BNE     DASCLP      NO, GO DO MORE
02610           LDA     #13         ELSE DO CARRIAGE RETURN
02620           JSR     CHROUT
02630           LDA     #10         AND LINE FEED
02640           JSR     CHROUT
02650           BRA     DASCLP      AND THEN MORE CHARACTERS
02660 DADONE    LDA     #13         256 BYTES DONE
02670           JSR     CHROUT      DO CARR RET
02680           LDA     #10         AND LINE FEED
02690           JSR     CHROUT
02700           CLR     $6F         RESTORE CODE TO SCREEN
02710           LBRA    MODLP       RET TO MODIFY ROUTINE
02720 DMPHEX    LBSR    PHEX        PRINT HEX DIGITS
02730           BRA     DALP2       GO FOR MORE
02740 *
02750 ************************************
02760 * INCREMENT SECTOR DATA FOR DSKCON *
02770 ************************************
02780 *
02790 INCRSC    LDA     SECTR,PCR   GET CURRENT SECTOR #
02800           INCA                INCREMENT IT
02810           CMPA    #19         18 SECTORS/TRACK ONLY
02820           BNE     INSOK       GO IF 18 OR LESS
02830           LDB     TRACK,PCR   GET CURRENT TRACK #
02840           INCB                INCREMENT TRACK
02850           CMPB    #35         TRACKS 0-34 ONLY
02860           BNE     INTOK       GO IT TRACK OK
02870           CLRB                ELSE MAKE TRACK 0
02880 INTOK     STB     TRACK,PCR   SAVE NEW TRACK #
02890           LDA     #1          MAKE SECTOR #1
02900 INSOK     STA     SECTR,PCR   SAVE NEW SECTOR #
02910           LBRA    START       RESTART
02920 *
02930 ************************************
02940 * DECREMENT SECTOR DATA FOR DSKCON *
02950 ************************************
02960 *
02970 DECRSC    LDA     SECTR,PCR   GET CURRENT SECTOR #
```

```
02980           DECA                    DECREMENT IT
02990           BNE       DCSOK         GO IF NOT ZERO
03000           LDB       TRACK,PCR     GET CURRENT TRACK #
03010           DECB                    DECREMENT TRACK
03020           CMPB      #$FF          LESS THAN ZERO?
03030           BNE       DCTOK         GO IF NOT
03040           CLRB                    MAKE TRACK 0
03050 DCTOK     STB       TRACK,PCR     SAVE NEW TRACK #
03060           LDA       #18           MAKE SECTOR #18
03070 DCSOK     STA       SECTR,PCR     SAVE NEW TRACK #
03080           LBRA      START         RESTART
03090 *
03100 *************************************
03110 * COMPUTE ADDRESS OF BYTE IN BUFFER *
03120 *************************************
03130 *
03140 COMPAD    LDD       $88           GET CURRENT CURSOR LOCATION
03150           SUBD      #$400         FIND OFFSET FROM START
03160           LSRA
03170           RORB
03180           LEAX      BFR,PCR       GET BUFFER ADRS INTO X
03190           LEAX      D,X           COMPUTE BFR ADRS OF BYTE
03200           RTS                     RETURN TO CALLER
03210 *
03220 *******************************
03230 * DUMP ASCII BUFFER TO SCREEN *
03240 *******************************
03250 *
03260 SCRASC    LDD       $88           GET CURRENT CURSOR LOCATION
03270           PSHS      D             SAVE IT
03280           LEAX      BFR,PCR       GET BUFFER ADRS
03290           LDY       #VIDEO        START OF VIDEO MEMORY
03300           LDB       #96           SPACE CHARACTER
03310 SALP      STB       ,Y+           CHAR TO SCREEN
03320           LDA       ,X+           GET CHARACTER
03330           ANDA      #$7F          CONVERT GRAPHICS CODES
03340           CMPA      #32
03350           BHS       SCRA1         GO IF PRINTABLE CODE
03360           LDA       #$2E          ELSE MAKE IT A PERIOD
03370 SCRA1     CMPA      #$40          CONVERT TO CORRECT CODE
03380           BHS       SCRA2
03390           ADDA      #$40
03400           BRA       SCRA3
03410 SCRA2     CMPA      #$60          CONVERT LOWER CASE TO UPPER
03420           BLO       SCRA3
03430           SUBA      #$60
03440 SCRA3     STA       ,Y+           PUT ON SCREEN
03450           CMPY      #$600         END OF SCREEN?
03460           BNE       SALP          LOOP IF NOT
03470           PULS      D             ELSE RECOVER CURSOR LOC
03480           ORB       #1
03490           STD       $88           & RESTORE IT
03500           CLR       MODE,PCR
03510           COM       MODE,PCR      SET ASCII MODE
03520           RTS                     AND RETURN TO CALLER
03530 *
03540 ****************************
03550 * DUMP HEX BUFFER TO SCREEN *
03560 ****************************
03570 *
03580 SCRHEX    LDD       $88           GET CURRENT CURSOR LOC
03590           PSHS      D             SAVE IT
03600           LEAX      BFR,PCR       GET BUFFER ADRS
03610           LDY       #VIDEO        POINT TO SCREEN START
03620 SHLP      STY       $88           CURSOR TO START
03630           LDA       ,X+           GET BYTE FROM BUFFER
03640           BSR       HEXCHR        PRINT IT (SPECIAL FORMAT)
03650           CMPY      #$600         END OF VIDEO?
03660           BNE       SHLP          REPEAT 'TIL DONE
03670           PULS      D             RECOVER CURSOR LOCATION
03680           ANDB      #$FE          MAKE EVEN ADRS
03690           STD       $88           RESTORE CURSOR
03700           CLR       MODE,PCR      SET HEX MODE
03710           RTS                     RETURN TO CALLER
03720 *
03730 ****************************
03740 * SPECIAL FORMAT HEX PRINT *
03750 ****************************
03760 *
03770 HEXCHR    PSHS      A             SAVE CHARACTER
03780           LSRA                    SHIFT IT
03790           LSRA
03800           LSRA
03810           LSRA
03820           BSR       HXCHR1        PRINT 1ST NIBBLE
03830           PULS      A             RECOVER ORIG CHAR
03840 HXCHR1    ANDA      #$0F          MASK OFF HIGH NIBBLE
03850           ADDA      #$30          MAKE ASCII DIGIT
03860           CMPA      #$3A          DIGIT A-F?
03870           BLO       CHRHEX        GO IF NOT
03880           ADDA      #7            ELSE CORRECT ASCII
03890 CHRHEX    LDB       $89           GET LOW BYTE OF CURS LOC
03900           ANDB      #2            IS IT EVEN OR ODD?
03910           BEQ       NONINV        NO INVERT IF EVEN BYTE
03920           CMPA      #$40          ALPHA CHARACTER?
03930           BLO       PUTHX         GO IF SO
03940           SUBA      #$40          ELSE INVERT NUMBER
03950 PUTHX     STA       ,Y+           STORE IN VIDEO MEMORY
03960           RTS                     RETURN TO CALLER
03970 NONINV    CMPA      #$40          ALPHA CHAR?
03980           BHI       PUTHX         GO IF SO
03990           ADDA      #$40          ELSE CONVERT DIGIT
04000           BRA       PUTHX         GO STORE DIGIT
04010 *
04020 *********************************
04030 * INPUT CHARACTER FROM KEYBOARD *
04040 * WITH BLINKING CHAR FOR CURSOR *
04050 *********************************
04060 *
04070 GETKEY    PSHS      B             SAVE B REG
```

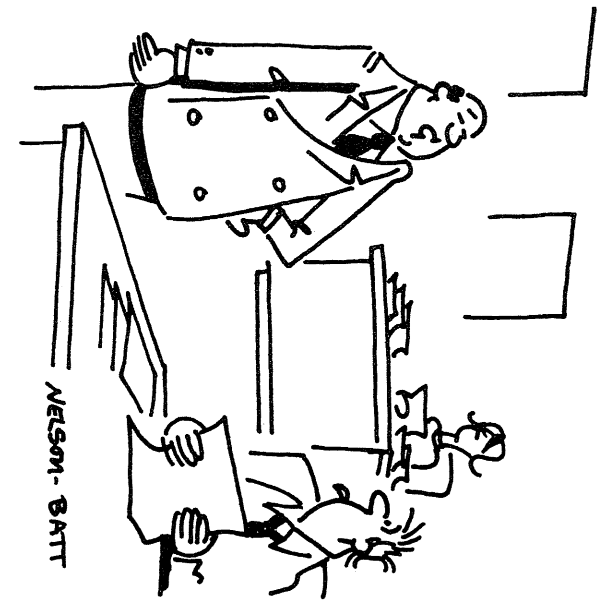
```
04080           LDA     [$88]       GET CHAR AT CURSOR LOC
04090           PSHS    A           SAVE IT
04100           CLRB                ZERO COUNTER
04110 GTKY1     JSR     KBSCAN      SEE IF KEY PRESSED
04120           BNE     GTKY2       GO IF KEY DOWN
04130           DECB                COUNT DOWN
04140           BNE     GTKY1       LOOP IF NOT TIMED OUT
04150           LDA     #$40        INVERT MASK
04160           EORA    [$88]       INVERT CURSOR CHAR
04170           STA     [$88]       PUT IT ON SCREEN
04180           BRA     GTKY1       REPEAT 'TIL KEY PRESSED
04190 GTKY2     PULS    B           GET ORIGINAL CHAR
04200           STB     [$88]       RESTORE IT TO SCREEN
04210           PULS    B,PC        RESTORE B AND RETURN
04220 *
04230 ***************************
04240 * CURSOR MOVEMENT ROUTINE *
04250 ***************************
04260 *
04270 CURS      PSHS    X           SAVE X REG
04280           LDX     $88         GET CURRENT CURSOR LOCATION
04290           CMPA    #8          LEFT ARROW?
04300           BEQ     LEFT
04310           CMPA    #9          RIGHT ARROW?
04320           BEQ     RIGHT
04330           CMPA    #10         DOWN ARROW?
04340           BEQ     DOWN
04350           CMPA    #94         UP ARROW?
04360           BEQ     UP
04370           ORCC    #1          NOT AN ARROW KEY, FLAG ERR
04380           BRA     CRSRT       AND CONTINUE
04390 CURSRT    ANDCC   #$FE        RETURN WITH NO ERROR
04400 CRSRT     STX     $88         UPDATE CURSOR LOCATION
04410           PULS    X,PC        RECOVER X AND RETURN
04420 LEFT      CMPX    #$400       UPPER LEFT CORNER?
04430           BEQ     CURSRT      RETURN IF SO
04440           LEAX    -1,X        ELSE DECREMENT CURSOR LOC
04450           TST     MODE,PCR    WHICH MODE?
04460           BEQ     CURSRT      GO IF HEX MODE
04470           LEAX    1,X         ELSE RESTORE CURS LOC
04480           CMPX    #$401       UPPER LEFT FOR ASCII?
04490           BEQ     CURSRT      GO IF SO
04500           LEAX    -2,X        ELSE DECR CURS LOC BY 2
04510           BRA     CURSRT      AND RETURN
04520 RIGHT     CMPX    #$5FF       LOWER RIGHT CORNER?
04530           BEQ     CURSRT      RETURN IF SO
04540           LEAX    1,X         ELSE INCR CURSOR LOCATION
04550           TST     MODE,PCR    WHICH MODE?
04560           BEQ     CURSRT      RETURN IF HEX MODE
04570           LEAX    1,X         ELSE INCR CURSOR AGAIN
04580           BRA     CURSRT      AND RETURN
04590 UP        CMPX    #$41F       UPPER EDGE OF SCREEN?
04600           BLS     CURSRT      RETURN IF SO
04610           LEAX    -32,X       ELSE MOVE CURSOR UP 1 LINE
04620           BRA     CURSRT      AND RETURN
04630 DOWN      CMPX    #$5E0       LOWER EDGE OF SCREEN?
04640           BHS     CURSRT      GO IF SO
04650           LEAX    32,X        ELSE MOVE CURS DOWN 1 LINE
04660           BRA     CURSRT      AND RETURN
04670 *
04680 *********************************
04690 * PRINT ASCII MESSAGE POINTED   *
04700 * TO BY X REG; TERM WITH 0 BYTE *
04710 *********************************
04720 *
04730 PRINT     LDA     ,X+         GET CHARACTER
04740           BEQ     PRTEND      GO IF ZERO BYTE
04750           JSR     CHROUT      ELSE PRINT IT
04760           BRA     PRINT       LOOP 'TIL DONE
04770 PRTEND    RTS                 RETURN TO CALLER
04780 *
04790 ****************************
04800 * PRINT HEX VALUE IN A REG *
04810 ****************************
04820 *
04830 PHEX      PSHS    A           SAVE VALUE
04840           LSRA                SHIFT HIGH NIBBLE
04850           LSRA
04860           LSRA
04870           LSRA
04880           BSR     PHEX1       PRINT HIGH NIBBLE
04890           PULS    A           RECOVER ORIG VALUE
04900 PHEX1     ANDA    #$0F        MASK OFF HIGH NIBBLE
04910           ADDA    #$30        MAKE ASCII
04920           CMPA    #$3A        ALPHA?
04930           BLO     PHEX2       GO IF NOT
04940           ADDA    #7          ELSE CORRECT ASCII
04950 PHEX2     JMP     CHROUT      GO PRINT IT
04960 *
04970 *******************************
04980 * INPUT 16 BIT HEX # TO X REG *
04990 *******************************
05000 *
05010 HEXIN     LDX     #0          INIT VALUE TO ZERO
05020           CLRB                B= # OF DIGITS INPUT
05030           PSHS    B,X         SAVE VALUES
05040 HEXIN1    LBSR    GETKEY      GET KEYBOARD INPUT
05050           CMPA    #13         IS IT ENTER?
05060           BNE     HEXIN2      GO IT NOT
05070           PULS    B,X,PC      RECOVER VALUES & RETURN
05080 HEXIN2    CLRB
05090           CMPA    #3          BREAK KEY?
05100           BEQ     HXBRK       GO IF SO
05110           CMPA    #8          BACKSPACE (LEFT ARROW)?
05120           BEQ     HXBAK       GO IF SO
05130           CMPA    #$30        ASCII '0'?
05140           BLO     HEXIN1      REPEAT IF NOT HEX
05150           CMPA    #$3A        ALPHA?
05160           BLO     HEXIN3      GO IF NOT
05170           CMPA    #$41        VALID HEX?
```

```
05180            BLO      HEXIN1      GO IF NOT
05190            CMPA     #$47
05200            BHS      HEXIN1
05210            JSR      CHROUT      ECHO CHAR TO SCREEN
05220            SUBA     #7          MAKE BINARY
05230            BRA      HEXIN3+3    & CONTINUE
05240 HEXIN3     JSR      CHROUT      ECHO DIGIT
05250            ANDA     #$0F        MAKE BINARY
05260            EXG      A,B         SWAP BYTES
05270            LDX      1,S         GET CURRENT VALUE
05280            EXG      D,X         SWAP VALUES
05290            LSLB                 DO A 16 BIT SHIFT
05300            ROLA
05310            LSLB
05320            ROLA
05330            LSLB
05340            ROLA
05350            LSLB
05360            ROLA
05370            STX      1,S         SAVE RESULT
05380            ADDD     1,S         ADD OLD VALUE
05390            STD      1,S         AND SAVE IT
05400            INC      0,S         INC DIGIT COUNT
05410            BRA      HEXIN1      & GO FOR MORE
05420 HXBAK      JSR      CHROUT      ECHO BACKSPACE
05430            LDD      1,S         GET OLD VALUE
05440            LSRA                 AND SHIFT IT BACK
05450            RORB
05460            LSRA
05470            RORB
05480            LSRA
05490            RORB
05500            LSRA
05510            RORB
05520            STD      1,S         RESTORE VALUE
05530            DEC      0,S         DECR DIGIT COUNT
05540            BRA      HEXIN1      & GO FOR MORE
05550 HXBRK      ORCC     #1          BREAK KEY HIT FLAG
05560            PULS     B,X,PC      RESTORE REGS & RETURN
05570 *
05580 *******************************
05590 * GET CHAR FROM KEYBOARD WITH *
05600 *     BLINKING DASH CURSOR    *
05610 *******************************
05620 *
05630 GETCHR     PSHS     B,X,Y       SAVE REGISTERS
05640 GCLP       LDA      #$6D        CURSOR CHAR (DASH)
05650            BSR      GETC        CHECK KEYBOARD
05660            BNE      GC2         GO IF KEY DOWN
05670            LDA      #$8F        SPACE CHARACTER
05680            BSR      GETC        CHECK KYBRD AGAIN
05690            BEQ      GCLP        LOOP IF NO KEY DOWN
05700 GC2        CMPA     #8          BACKSPACE?
05710            BNE      GCP         GO IF NOT
05720            LDA      #$8F        SPACE CHAR
05730            STA      [$88]       TO CURSOR LOCATION
05740            LDA      #8          RESTORE LEFT ARROW
05750 GCP        PULS     PC,B,X,Y    RECOVER REGS & RETURN
05760 GETC       STA      [$88]       PUT CURS CHAR ON SCREEN
05770 GC4        JSR      KBSCAN      KEY PRESSED?
05780            BNE      GC3         RET IF SO
05790            DECB                 ELSE DEC COUNTER
05800            BNE      GC4         & TRY AGAIN
05810 GC3        RTS                  OR RETURN TO CALLER
05820 *
05830 *****************************
05840 *      ASCII MESSAGES       *
05850 *****************************
05860 *
05870 MSG        FCC      '*********  CCZAP V1.1  *********'
05880            FCC      '**** (C) 1983 BY J. L. POST ****'
05890            FCC      '            >>MENU<<'  (12 SPACES)
05900            FCB      13
05910            FCB      13
05920            FCC      'R =READ SECTOR  W =WRITE SECTOR'
05930            FCB      13
05940            FCC      'M =MODIFY DATA  B =GO TO BASIC'
05950            FCB      13
05960            FCC      '+ =INC SECTOR   - =DEC SECTOR'
05970            FCB      13
05980            FCC      '<BREAK>   RETURN TO MENU'
05990            FCB      13
06000            FCC      '<CLEAR>   TOGGLE HEX/ASCII'
06010            FCB      13
06020            FCC      '<ARROWS>  POSITION CURSOR'
06030            FCB      13
06040            FCC      '<SHFT-CLR> DUMP TO PRINTER'
06050            FDB      $0D0D
06060            FCC      '           COMMAND?'  (11 SPACES)
06070            FCB      13
06080            FCC      '           DRIVE #'
06090            FCB      13
06100            FCC      '           TRACK #'
06110            FCB      13
06120            FCC      '           SECTOR #'
06130            FCB      0
06140 ERMSG      FCC      'I/O ERROR'
06150            FCB      0
06160 LPTRK      FCB      13
06170            FCB      10
06180            FCC      'TRACK #'
06190            FCB      0
06200 LPSEC      FCC      '   SECTOR #'
06210            FCB      0
06220 TRACK      FCB      0
06230 SECTR      FCB      1
06240 MODE       FCB      0
06250 DRIVE      FCB      0
06260 ERFLG      FCB      0
06270 BFR        FCB      0
06280            END      START
```

Computers: Power Tools for Education

Continued from page 101

```
1 '************************************
  *   NUMBERS (FOR PREPRIMARY ED)      *
2 '*               by                  *
  *           M J MERCHANT             *
3 '*    LOS ANGELES, CA  NOV.1983      *
4 '************************************

5 CLEAR500
10 CLS:PRINT@32," YOU MUST REMOVE THE PROGRAM"," CASSETTE AND INSERT THE "," 60 SECOND ENDL
ESS-LOOP TAPE",
20 PRINT" LABELED 'NUMBER SOUNDS'":PRINT:PRINT" PRESS <PLAY> ON THE RECORDER"
30 PRINT@448," ANY COMPUTER KEY WILL CONTINUE"
40 A$=INKEY$:IFA$=""THEN40
50 CLS0
60 P=224
70 FORR=0TO4
80 FORI=0TO31
90 READT
100 PRINT@P,CHR$(159+T);
110 IFI/5=INT(I/5)THENSOUND20,1:SOUND200,1
120 P=P+1
130 NEXTI,R

133 REM LO RES "NUMBERS"

135 DATA16,11,1,16,1,16,1,1,16,1,16,11,1,1,16,11,6,16,16,12,1,16,16,16,1,16,16,12,1,8,16,16
145 DATA16,16,1,16,1,16,1,1,16,1,16,16,12,16,16,11,6,11,1,16,1,16,1,1,1,16,1,16,1,16,1,1
155 DATA16,6,11,16,1,16,1,1,16,1,16,1,16,11,6,11,6;16,16,12,1,16,16,16,1,16,16,12,1,14,16,1
2
165 DATA16,1,16,16,1,16,1,1,16,1,16,1,1,1,6,11,6,11,1,16,1,16,1,1,1,16,1,16,1,3,1,16
175 DATA16,1,6,16,1,6,16,16,11,1,16,1,1,1,6,11,6,16,16,15,1,16,16,16,1,16,1,16,1,16,16,16
190 FORS=0TO13:SOUNDRND(18),1:FORQ=0TO120:NEXTQ,S
200 RESTORE
210 POKE65494,0':A=33:C=127
220 PMODE3,1
230 CLS
240 PRINT" CHOOSE ANY UNMARKED NUMBER."
250 PRINT@128," WHICH NUMBER 1 2 3 4 5 6 7 8 9"
255 L=N
260 PRINT@173,ND$;:ND$=""
270 FORI=0TOLL:LL$=LL$+" ":NEXT
280 PRINT@168,LL$;
290 LL$=""
300 N$=INKEY$:IFN$=""THENQ=Q+1:IFQ/120=INT(Q/120)THENSOUND240,1
310 IFN$=""THEN300ELSEN=VAL(N$)
320 IFN=>L-3ANDN<L+1ORN>5ANDL=0  THEN300
330 IFASC(N$)>57ORASC(N$)<49THEN300
340 FORI=1TO N:ND$=ND$+" *":NEXT
350 SOUND60,5:FORQ=0TO30:NEXT:SOUND130,5:FORQ=0TO60:NEXT:SOUND160,6
360 CLS(N-1)
370 PRINT@176," "N$" ";

375 REM LOCATE SPOT ON TAPE LOOP

380 OPEN"I",-1,N$
390 IFEOF(-1)THENCLOSE-1

393 REM PLAY MESSAGE

395 POKE65495,0
400 AUDIOON:MOTORON
410 PRINT@326," YOUR NUMBER IS ";N" ";
420 FORQ=0TO1300:NEXTQ
430 MOTOROFF
435 POKE65494,0
440 FORQ=0TO400:NEXTQ:FORI=0TO (N+158):READN$:NEXT
445 CC=RND(3)
450 IFN=0THENCLS0:FORQ=0TO400:NEXT:GOTO250
460 READN$
470 PMODE3,1:PCLS2:COLOR1,2
480 SCREEN1,1
490 PCLS(CC+1)
500 DRAWN$:FORQ=0TO300:NEXT
505 POKE65495,0
510 PAINT(28,190),CC+2,1:PAINT(158,60),CC+2,1:PAINT(130,110),CC+2,1:SOUND222,1
515 POKE65494,0
520 PMODE4,1:SCREEN1,1
530 FORQ=0TO200:NEXT

535 REM NUMBERS ONE - NINE

540 DATA"BM120,90;S18U4E4R4D20R4D4L12U4R4U12L4"
550 DATA"BM110,90;S22U5E4R8F4D7G11R11D4L16U3E13U5H3L4G3D4L3"
560 DATA"BM110,40;S26R15D5G4F4D5G4L8H4U3R2D2F3R6E2U5H3L2U2R2E4U2L13U2"
570 DATA"BM82,118;S40L2E10R3D10R3D2L3D5L3U5L10U2R10U7G7"
580 DATA"BM120,90;S36U8R13D2L11D4R9F3D6G3L8H3U3R2D2F3R4E3U4H2L10"
590 DATA"BM120,82;S39R7F3D3G3L6H3U10E3R6F3D3L2U3H2L4G2D4BD2R6F2D2G2L4H2U4"
600 DATA"BM119,89;S30U3R13D4M-12,+13U4M+9,-9U1L10"
610 DATA"SM120,90;S29H2U3E3R4F3D3G2F2D3G3L4H3U3E2BR2H2U2E2R2F2D2G2F2D2G2L2H2U2E2"
620 DATA"BM120,120;S37D2F3R6E3U10H3L6G3D3F3R7D4G2L4H2U2L2BR10BU4U3H2L4G2D1F2R6"
630 FORI=0TO36:SOUNDRND(255),RND(3):FORQ=0TORND(6):NEXTQ,I
640 T=1:S=1:Z=0
650 RESTORE
660 IFN=3THENN=0:L=6:LL=14:LL$="":GOTO680
670 LL=(2*N)-4
680 FORI=0TO6:T=T+.27:SOUND110,T:FORQ=0TOZ:NEXTQ:Z=Z+16:NEXTI
690 FORQ=0TO1200:NEXT:GOTO210
```



WHILE I'M ON VACATION, THE COMPUTER WILL BE IN CHARGE,

AD INDEX

*Manufacturer requests factory-direct inquiry. This index is provided as an additional service. The publisher does not assume any liability for errors or omissions.